# APA NEWSLETTER ON

# Philosophy and Computers

Ange Cooksey and Peter Boltuć, Co-Editors Spring 2007 Volume 06, Number 2

## FROM THE ISSUE EDITOR

**Peter Bołtuć**
*University of Illinois–Springfield*

Philosophy and computers is no longer a new, controversial, or unproven field; it is now mainstream philosophy, and it is the goal of the present and future issues of this *Newsletter* to make this fact conspicuous. As mainstream philosophy, philosophy and computers touches on two crucial areas of our discipline: first, it shares many issues with philosophy of mind and cognitive science. Second, it deals with moral and political philosophy as applied to computers and their rapid development. Both areas are represented in this *Newsletter*.

We may have at least a couple of history-making articles in the current issue. We open with Gilbert Harman's defense of first-person non-reductionism; the view has been presented by Harman in print twice in the last decade or so, but this may be the clearest expression yet. The relevance of Harman's paper to philosophy and computers should be obvious; yet, it is made even more so by the following two or three articles.

William Rapaport argues three points that belong both to philosophy and computers and to philosophy of mind: that it is an empirical question whether the brain is a digital computer; that the mind is a computer program; and that not only is it true that operations of the brain can be simulated on a digital computer, but also non-trivially so. Rapaport's discussion includes a number of interesting points, such as criticism of Searle's distinction among strong AI, weak AI, and cognitivism.

The international perspective is represented in this issue by Gordana Dodig-Crnkovic from Sweden. The author introduces computationalist epistemology as a broad, nearly metaphysical, explanatory framework. The article follows up on a paper Dodig-Crnkovic gave at the APA's Eastern Division meeting in December 2005 as a part of a panel organized by the Committee on Philosophy and Computers.

James Moor's article may serve as a link between epistemological issues and computer ethics. The author presents a powerful defense of the view that robot ethics is a legitimate and important field of research. Moor argues that robots should be viewed as explicit ethical agents—less than full ethical agents but more than implicit ethical agents (for definitions of those terms the reader is encouraged to consult the article). It is our particular pleasure to publish Moor's paper in this issue since the author is the recent winner of the Barwise Prize awarded by the Committee; more on this in the *Note from the Chair*.

Keith Miller's article discusses the question of moral responsibility of software developers for software uses by others after its release, with particular focus on open-source software. The paper is built on an extensive study of a recent case pertaining to an attempt to forbid military uses of certain kinds of open-source software. Miller discusses this case in the context of broader open-source philosophy. A version of this paper was presented at the APA's Eastern Division meeting in December 2006 (at the panel organized by the International Association for Computing and Philosophy).

The second part of this issue is devoted to the Committee business and to the call for papers. It features the *Note from the Chair*, by Marvin Croy. Despite the recent change of submission deadline by the APA to January 2nd, the Chair was able to give us not only an up-to-date overview of the committee's activities but also photographs of the awarding of the Barwise Prize to James Moor that took place a few days earlier. I would like to note the recent change in the structure of the Committee with the addition of an associate chair, and to welcome Michael Byron in that role. The associate chair will become the chair on July 1st, 2007.

I would also like to alert readers to the opportunity of participating at the 2007 North American Meeting of the International Association for Computing and Philosophy (NA-CAP) to take place in Chicago on July 26-28. The conference covers the full spectrum of issues in philosophy and computers, from distance learning and electronic teaching and research resources, through computer ethics, social and metaphysical issues, to computational logic and philosophical issues in cognitive science, AI, and robotics.

The current issue of the *Newsletter* consists of two parts. The first part features five articles: the first three pertain to philosophy and computers in the context of epistemology and philosophy of mind, while the other two pertain primarily to ethics. It is a good idea to aim at having about this number of papers on philosophy and computers in every issue, distributed between core and moral/political philosophy. The second part is devoted to current issues in the work on the Committee, prominently the note from the chair, calls for papers, and other housekeeping issues. In future issues this part of the *Newsletter* will need to become much more of a community project incorporating discussions and information on various events pertaining to the charges of this Committee. I would also like to revive a fruitful practice from an earlier period of this *Newsletter*: it would help to have short discussions of interesting websites, and perhaps even computer programs or games, that are of special interest to philosophers. Let me finish with a call for articles (these will remain peer-reviewed), discussion notes (especially those referring to the earlier articles and notes published here), reviews, invitations, other contributions, and, last but not least, for more pictures.

I would like to thank the Committee for providing me with the opportunity of editing this issue of the *Newsletter* and Margot Duley, the Dean of Liberal Arts and Sciences at the University of Illinois (Springfield campus), for making it possible for me to devote some time and attention to this task. Last but not least, I want to thank my Graduate Assistants, Charles Byrne and Joy Waranyuwat, from the University of Illinois (Urbana campus), for their extensive assistance, and to give very special thanks to the various philosophers all over the U.S. for serving as reviewers.

# ARTICLES

## *Explaining an Explanatory Gap*[1]

**Gilbert Harman**
*Princeton University*

Discussions of the mind-body problem often refer to an "explanatory gap" (Levine 1983) between some aspect of our conscious mental life and any imaginable objective physical explanation of that aspect. It seems that whatever physical account of a subjective conscious experience that we might imagine will leave it completely puzzling why there should be such a connection between the objective physical story and the subjective conscious experience (Nagel 1974).

What is the significance of this gap in understanding? Chalmers (1996) takes the existence of the gap to be evidence against physicalism in favor of some sort of dualism. Nagel (1974) and Jackson (1982, 1986) see it as evidence that objective physical explanations cannot account for the intrinsic quality of experience, although Jackson (1995, 1998, 2004) later changes his mind and comes to deny that there is such a gap. Searle (1984) argues that an analogous gap between the intrinsic intentional content of a thought or experience and any imaginable functionalist account of that content is evidence against a functionalist account of the intrinsic intentionality of thoughts. On the other hand, McGinn (1991) suggests that these explanatory gaps are due to limitations on the powers of human understanding—we are just not smart enough!

A somewhat different explanation of the explanatory gap appeals to a difference, stressed by Dilthey (1883/1989) and also by Nagel (1974), between two kinds of understanding, objective and subjective. Objective understanding is characteristic of the physical sciences—physics, chemistry, biology, geology, and so on. Subjective understanding does not play a role in the physical sciences but does figure in ordinary psychological interpretation and in what Dilthey calls the "Geisteswissenschaften"—sciences of the mind broadly understood to include parts of sociology, economics, political theory, anthropology, literary criticism, history, and psychology, as well as ordinary psychological reflection.

The physical sciences approach things objectively, describing what objects are made of, how they work, and what their functions are. These sciences aim to discover laws and other regularities involving things and their parts, in this way achieving an understanding of phenomena "from the outside." The social and psychological sciences are concerned in part with such objective understanding, but admit also of a different sort of subjective understanding "from the inside," which Dilthey calls "Das Verstehen." Such phenomena can have content or meaning of a sort that cannot be appreciated within an entirely objective approach. There are aspects of reasons, purposes, feelings, thoughts, and experiences that can only be understood from within, via sympathy or empathy or other translation into one's own experience.

Suppose, for example, we discover the following regularity in the behavior of members of a particular social group. Every morning at the same time each member of the group performs a fixed sequence of actions: first standing on tip toe, then turning east while rapidly raising his or her arms, then turning north while looking down, and so on, all this for several minutes. We can certainly discover that there is this objective regularity and be able accurately to predict that these people will repeat it every morning, without having any subjective understanding of what they are doing—without knowing whether it is a moderate form of calisthenics, a religious ritual, a dance, or something else. Subjectively to understand what they are doing, we have to know what meaning their actions have for them. That is, not just to see the actions as instances of an objective regularity.

Similarly, consider an objective account of what is going on when another creature has an experience. Such an account may provide a functional account of the person's brain along with connections between brain events and other happenings in the person's body as well as happenings outside the person's body. Dilthey and later Nagel argue that a completely objective account of a creature's experience may not itself be enough to allow one to understand it in the sense of being able to interpret it or translate it in terms one understands in order to know what it is like for that creature to have that experience. Such an account does not yet provide a translation from that creature's subjective experience into something one can understand from the inside, based on one's own way of thinking and feeling.

Nagel observes that there may be no such translation from certain aspects of the other creature's experiences into possible aspects of one's own experiences. As a result, it may be impossible for a human being to understand what it is like to be a bat.

We are not to think of Das Verstehen as a method of discovery or a method of confirming or testing hypotheses that have already been formulated. It is rather needed in order to understand certain hypotheses in the first place. So, for example, to understand a hypothesis or theory about pain involves understanding what it is like to feel pain. An objective account of pain may be found in biology, neuroscience, and psychology, indicating, for example, how pain is caused and what things pain causes (e.g., Melzack and Wall 1983). But it is possible to completely understand this objective story without knowing what it is like to feel pain. There are unfortunate souls who do not feel pain and are therefore not alerted to tissue damage by pain feelings of burning or other injury (Cohen, Kipnis, Kunkle, and Kubsansky 1955). If such a person is strictly protected by anxious parents, receives a college education, and becomes a neuroscientist, could that person come to learn all there is to learn about pain? It seems that such a person might fail to learn the most important thing—what it is like to experience pain—because objective science cannot by itself provide that subjective understanding.

Recent defenders of the need for Das Verstehen often mention examples using color or other sensory modalities, for example, a person blind from birth who knows everything there is to know from an objective standpoint about color and people's perception of it without knowing what red things look like to a normal observer (Nagel 1974).

With respect to pain and other sensory experiences there is a contrast between an objective understanding and a subjective understanding of what it is like to have that experience, where such a subjective understanding involves seeing how the objective experience as described from the outside translates into an experience one understands from the inside.

In thinking about this, I find it useful to consider an analogous distinction in philosophical semantics between accounts of meaning in terms of objective features of use, for example, and translational accounts of meaning.

For Quine (1960), an adequate account of the meaning of sentences or other expressions used by a certain person or group of people should provide translations of those expressions into one's "home language." In this sort of view, to give the meaning of an expression in another language is to provide a synonymous expression in one's own language. Similarly, if one wants to give the content of somebody else's thought, one has to find a concept or idea of one's own that is equivalent to it.

Imagine that we have a purely objective theory about what makes an expression in one's home language a good translation of an expression used by someone else. For example, perhaps such a theory explains an objective notion of use or function such that what makes one notion the correct translation of another is that the two notions are used or function in the same way. Such a theory would provide an objective account of correct translation between two languages, objectively described. (This is just an example. The argument is meant to apply to any objective account of meaning.)

To use an objective account of translation to understand an expression as used in another language, at least two further things are required. First, one must be able to identify a certain objectively described language as one's own language, an identification that is itself not fully objective. Second, one must have in one's own language some expression that is used in something like the same way as the expression in the other language. In that case, there will be an objective relation of correct translation from the other language to one's own language, which translates the other expression as a certain expression in one's own language. Given that the correct translation of the other expression is an expression in one's own language "E," one can understand that the other expression means E. "Yes, I see, 'Nicht' in German means not."

This is on the assumption that one has an expression "E" in one's own language that correctly translates the expression in the other language. If not, Das Verstehen will fail. There will be no way in one's own language correctly to say or think that the other expression means E. There is no way to do it except by expanding the expressive power of one's language so that there is a relevant expression "E" in one's modified language.

Let me apply these thoughts about language to the more general problem of understanding what it is like for another creature to have a certain experience. Suppose we have a completely objective account of translation from the possible experiences of one creature to those of another, an account in terms of objective functional relations, for example. That can be used in order to discover what it is like for another creature to have a certain objectively described experience given the satisfaction of two analogous requirements. First, one must be able to identify one objectively described conceptual system as one's own. Second, one must have in that system something with the same or similar functional properties as the given experience. To understand what it is like for the other creature to have that experience is to understand which possible experience of one's own is its translation.

If the latter condition is not satisfied, there will be no way for one to understand what it is like to have the experience in question. There will be no way to do it unless one is somehow able to expand one's own conceptual and experiential resources so that one will be able to have something corresponding to the other's experience.

Knowledge that P requires being able to represent its being the case that P. Limits on what can be represented are limits on what can be known. If understanding what it is like to have a given experience is an instance of knowing that something is the case, then lacking an ability to represent that P keeps one from knowing that something is the case.

About the case in which nothing in one's own system could serve to translate from another creature's experience to one's own, Nemirov (1980), Lewis (1998), and Jackson (2004) say in effect that one might merely lack an ability, or know-how, without lacking any knowledge that something is the case. For them, understanding what it is like to have a given experience is not an instance of knowing that something is the case, a conclusion that I find bizarre.

I prefer to repeat that a purely objective account of conscious experience cannot always by itself give an understanding of what it is like to have that experience. There will at least sometimes be an explanatory gap. This explanatory gap has no obvious metaphysical implications. It reflects the distinction between two kinds of understanding: objective understanding and Das Verstehen.

### Endnotes

1. For additional discussion, see (Harman 1990,1993). I am indebted to comments from Peter Boltuć and Mary Kate McGowan.

### Bibliography

Chalmers, D. *The Conscious Mind*. Oxford, NY: Oxford University Press, 1996.

Cohen, L. D., D. Kipnis, E. C. Kunkle, and P. E. Kubsansky. "Case report: Observation of a person with congenital insensitivity to pain." *Journal of Abnormal and Social Psychology* 51 (1955): 333-38.

Dilthey, W. *Introduction to the Human Sciences*, edited by R. Makkreel and F. Rodi. Princeton, NJ: Princeton University Press, 1989. Original work published 1883.

Harman, G. "Immanent and Transcendent Approaches to Meaning and Mind." In *Perspectives on Quine*, edited by R. Gibson and R. B. Barrett. Oxford: Blackwell, 1990; reprinted in G. Harman, Reasoning, Meaning, and Mind. 262-75. Oxford: Oxford University Press, 1999.

Harman, G. "Can Science Understand the Mind?" In *Conceptions of the Human Mind: Essays in Honor of George A. Miller*, edited by G. Harman. Hillsdale, NJ: Lawrence Erlbaum, 1993.

Jackson, F. "Epiphenomenal Qualia." *Philosophical Quarterly* 32 (1982): 127-32.

Jackson, F. "What Mary Didn't Know." *Journal of Philosophy* 83 (1986): 291-95.

Jackson, F. "Postscript." In *Contemporary Materialism*, edited by P. Moser and J. Trout. London: Routledge, 1995.

Jackson, F. "Postscript on Qualia." In his *Mind, Method, and Conditionals*. London: Routledge, 1998.

Jackson, F. "Mind and Illusion." P. Ludlow, Y. Nagasawa, and D. Stoljar. *There's Something About Mary: Essays on Phenomenal Consciousness and Frank Jackson's Knowledge Argument*. Cambridge, MA: MIT Press, 2004.

Levine, J. "Materialism and Qualia: The Explanatory Gap." *Pacific Philosophical Quarterly* 64 (1983): 354-61.

Lewis, D. *Proceedings of the Russellian Society*, edited by J. Copley-Coltheart. 13 (1988): 29-57.

Melzack, R., and P. D. Wall. *The Challenge of Pain* (rev. ed.). New York: Basic Books, 1983.

Nagel, T. "What Is It Like to be a Bat?" *Philosophical Review* 83 (1974): 435-50.

Nemirov, L. "Review of T. Nagel, Moral Questions." *Philosophical Review* 89 (1980): 475-76.

Quine, W. V. *Word and Object*. Cambridge, MA: MIT Press, 1960.

Searle, J. *Minds, Brains, and Science*. Cambridge, MA: Harvard University Press, 1984.

## Searle on Brains as Computers

**William J. Rapaport**
*State University of New York at Buffalo*

### Abstract

Is the brain a digital computer? Searle says that this is meaningless; I say that it is an empirical question. Is the mind a computer program? Searle says no; I say: properly understood, yes. Can the operations of the brain be simulated on a digital computer? Searle says: trivially yes; I say yes, but that it is not trivial.

### 1. Three Questions

In his 1990 essay "Is the Brain a Digital Computer?" Searle factors the "slogan...'the mind is to the brain as the program is to the hardware'" (p. 21) into three questions (Searle 1990, 21):

1. Is the brain a digital computer?
2. Is the mind a computer program?
3. Can the operations of the brain be simulated on a digital computer?

Let us consider each of these, beginning with the second.

### 2. Is the Mind a Computer Program?

What does it mean to say that the mind is a computer program? Surely not that there is a programming language and a program written in it that is being executed on a brain—not for humans, at least. But it could mean that by bottom-up, reverse engineering (neuroscience) together with top-down, cognitive-scientific investigation, we could write a program that would cause a computer to exhibit mental behavior. However, that's question 3, to which Searle gives a different answer.

Possibly, question 2 means that the mind plays the same role with respect to the brain that a program does to a computer; call this "Good Old-Fashioned Cartesian Dualism." This may not be much progress over the "slogan" of which question 2 is supposed to be merely a part, but it is worth a small digression.

### 2.1 Good Old-Fashioned Cartesian Dualism

Computational cognitive science, including what John Haugeland (1985, 112) has termed "good old-fashioned artificial intelligence," is, I believe, good old-fashioned Cartesian dualism. The view that mental states and processes are (or are expressible as) algorithms that are *implemented in* the physical states and processes of physical devices is (a form of) Cartesian dualism: The mental states and processes and the physical states and processes can be thought of as different "substances" that "interact." How might this be?

It should be clear that an algorithm and a computer are different kinds of "substance." If one considers an algorithm as a mathematical abstraction (in the ordinary sense of the term "abstraction"), then it is an abstract mathematical entity (like numbers, sets, etc.). Alternatively, if one considers an algorithm as a text expressed in some language, then it is, say, ink marks on paper or ASCII characters in a word processor's file. An algorithm might even be—and indeed ultimately is—"switch settings" (or their electronic counterparts) in a computer. All of these are very different sorts of things from a very physical computer.

How do an algorithm and a computer "interact"? By the latter being a semantic interpretation—a model—of the former. More precisely, the *processes* of the brain/body/computer are semantic interpretations (or models) of the mind/algorithm in the sense of semantics as correspondence. But this is just what we call an implementation. So, an implementation is a kind of semantic interpretation. (For further discussion of this, see Rapaport 1999, 2005.)

Note, by the way, that the mind/algorithm can itself be viewed as a semantic interpretation of the brain/body/computer, since the correspondence can go both ways (Rapaport 2002). How is a mind implemented? Consider a computer program: ultimately, the program (as text) is implemented as states of a computer (expressed in binary states of certain of its components). That is purely physical, but it is *also* purely syntactic; hence, *it* can have a semantic interpretation. An abstract data type, for instance, can be thought of as the semantic interpretation of an arrangement of bits in the computer (cf. Tenenbaum & Augenstein 1981, 1, 6, 45; see Rapaport 1995, §2.3, and Rapaport 2005). This Janus-faced aspect of the bit arrangements—thought of both as a physical model or implementation of the abstract algorithm and as a syntactic domain interpretable by the algorithm and its data structures—is Marx W. Wartofsky's "model muddle" (Wartofsky 1966, 1979, xiii–xxvi; Rapaport 1995; 1996, Ch. 2; 1999; 2000).

Now, is this really good old-fashioned Cartesian dualism? Is mind-body interaction really semantic interpretation or implementation? Or might this semantic/implementational view be more like some other theory of the mind?

It is not parallelism, since there really is a *causal* interaction: the algorithm (better: the process) *causes* the physical device to behave in certain ways.

So it's not epiphenomenalism, either. Moreover, the device—or its behavior—can produce changes in the program, as in the case of self-modifying programs, or even in the case of a system competent in natural language whose knowledge base (part of the software) changes with each interaction.

Could it be a dual-aspect theory? Perhaps: certainly, the physical states and processes are one "level of description," and the mental states and processes are another "level of description" *of the same (physical) system*. But talk of levels of description seems to me to be less illuminating than the theory of semantics as correspondence. More to the point, neither "level" is a complete description of the system: the algorithm is not the process, nor can one infer from the algorithm what the future behavior of the process will be: the process can behave in ways not predictable by the programmer (cf. Fetzer 1988, 1991). And even a complete physical description of the system would not tell us *what* it is doing; this is one of the lessons of functionalism.

So dualism is at least plausible. Do the physical states and processes produce mental ones? Here is where the problem of qualia—i.e., subjective qualitative experiences, including pain and physical sensations—enters. (I say more about this in Rapaport 2005, §2.2.)

### 2.2 Return to Searle

Does question 2 mean that the mind is the *way* the brain behaves? That seems right but isn't the right analogy: it *doesn't* seem right to say that a program is the way a computer behaves.

"Programs," Searle goes on to say, "are defined purely formally or syntactically" (p. 21). That, I think, is not *quite* right: they require a set of input-output conventions, which would be "links" to the world. In any case, this together with the assertion that "minds have an intrinsic content...immediately [implies] that the program by itself cannot constitute the mind" (p. 21). What does "content" mean?

If it means something internal to the mind (a "container" metaphor; cf. Twardowski 1894, Rapaport 1978), then that minds have intrinsic content could mean that within a mind there are links among mental concepts, some of which play the role of a language of thought and others of which play the role of mental representations of external perceptions (cf. Rapaport 1996, Ch. 3; 2000; 2002; 2006). If so, that would be purely syntactic, as Searle says programs are.

If, on the other hand, "content" means a relation to an external entity, then why don't programs have that, too (as I noted two paragraphs back)? In any case, programs do take input from the external world: I enter "2" on the keyboard, which results (after a few transductions) in a switch being set in the computer, which the program interprets as the number 2.

So, on either interpretation, the conclusion doesn't follow, since programs can *also* have "intrinsic mental content," whatever that means.

The problem is that question 2 is not the right question. Of course "the formal syntax of the program does not by itself guarantee the presence of mental contents" (p. 26) because the program might never be executed. What Searle should have asked is whether the mind is a computer *process*. And here the answer can be "yes," since *processes* can have contents. Searle says:

> I showed this [viz., that the formal syntax of a program doesn't guarantee the presence of mental contents] a decade ago in the Chinese Room Argument. …The argument rests on the simple logical truth that syntax is not the same as, nor is it by itself sufficient for, semantics. (Searle 1990, 21)

This seems to follow from Charles Morris's definitions (1938) of syntax as the study of relations *among* symbols and of semantics as the study of relations *between* symbols and their meanings; thus, syntax … semantics. Nor is it the case that semantics can be "derived," "constructed," or "produced" from syntax by Morris's definitions. But the first-person semantic enterprise *is* one of determining correspondences among symbols—between linguistic symbols and internal representations of external objects. Hence, it *is* syntactic even on Morris's definition. The *third*-person semantic enterprise is more like what Morris had in mind. But one person's third-person semantics is another's first-person semantics: if one cognitive agent, Oscar, tries to account for the semantics of another cognitive agent (Cassie) by drawing correspondences between her mental concepts and things in the world, all he can really do is draw correspondences between his representations of her concepts and his representations of things in the world. As with the turtles who hold up the Earth, it's syntax all the way down. (For more about how Cassie and Oscar understand each other, see, e.g., Rapaport 2003.)

### 3. Can the Operations of the Brain Be Simulated on a Digital Computer?

Let's turn to question 3, the answer to which Searle thinks is trivially—or, at least, uninterestingly—affirmative. "[N]aturally interpreted, the question means: Is there some description of the brain such that under that description you could do a computational simulation of the operations of the brain" (p. 21). Such a description would inevitably be partial (Smith 1985). Hence, so would be the computational simulation. But if it passed the Turing test (i.e., if its effects in the actual world were indistinguishable from those of a human), then what's not in the model is an implementation detail. What might these be? They might include sensations of pain, warm fuzzy feelings associated with categorizing something as "beautiful," etc. (cf. Rapaport 2005). As for pain, don't forget that our *sensation* of it is an internal perception, just like our sensation of an odor (cf. Crane 1998). It might be possible to be in pain and to know that one is in pain without what we normally call a pain sensation, just as it is possible to determine the presence of an object by its odor—by a chemical analysis—without sensing that odor.[1] The "triviality" or "obviousness" of the answer to question 3 stems, according to Searle, from Church's Thesis: "The operations of the brain can be simulated on a digital computer in the same sense in which weather systems, the behavior of the New York Stock market or the pattern of airline flights over Latin America can" (p. 21). And, presumably, since simulated weather isn't weather, simulated brains aren't brains. But the premise is arguable (Rapaport 2005, §3); at least, it does not follow that the behavior of simulated brains isn't *mental*. Brains and brain behavior are special cases.

### 4. Is the Brain a Digital Computer?

Searle equates question 1 to "Are brain processes computational?" (p. 22). What would it mean to say that the brain was *not* a digital computer? It might mean that the brain is *more* than a digital computer—that only some proper *part* of it *is* a digital computer. What would the rest of it be? Implementation details, perhaps. I am, however, willing to admit that perhaps not all of the brain's processes are computational. Following Philip N. Johnson-Laird (1988, 26–27), I take the task of cognitive science to be to find out *how much* of the brain's processes *are* computational—and surely some of them are (Rapaport 1998). It is, thus, a working hypothesis that brain processes are computational, requiring an empirical answer and not subject to a priori refutation.

On the other hand, to say that the brain is not a digital computer might mean that it's a different kind of entity altogether—that no part of it is a digital computer. But that seems wrong, since it *can* execute programs (we use our brains to hand-simulate computer programs, which, incidentally, is the inverse of Turing's 1936 analysis of computation).

What are brain processes, how do they differ from mental processes, and how do both of these relate to computer processes? A computer process is a program being executed; therefore, it is a physical thing that implements an abstract program. A brain process is also a physical thing, so it might correspond to a computer process. A mental process could be either (i) something abstract yet dynamic or (ii) a brain process. The former (i) makes no sense if programs and minds are viewed as static entities. The latter (ii) would mean that *some* brain processes are mental (others, like raising one's arm, are not). So to ask if brain processes are computational is like asking if a computer process is computational. That question means: Is the current behavior of the computer describable by a recursive function (or is it just a fuse blowing)? So Searle's question 1 is: Is the current (mental) behavior of the brain describable by a recursive function? This is the fundamental question of artificial intelligence as computational philosophy. It is a major research program, not a logical puzzle capable of a priori resolution.

Searle's categorization of the possible positions into "strong AI" ("all there is to having a mind is having a program"), "weak AI" ("brain processes [and mental processes] can be simulated computationally"), and "Cognitivism" ("the brain is a digital computer") is too coarse (p. 22). What about the claim that a computer running the "final" AI program (the one that passes the Turing test, let's say) has mentality? As I argued above, that's not necessarily "just" having a program. But on the *process* interpretation of question 2, Strong AI could be the view that all there is to having a mind is having a *process*, and that's more than having a program. What about the claim that the "final" AI program need not be the one that humans use—i.e., the claim that computational *philosophy* might "succeed," not computational *psychology*? (Computational philosophy seeks to learn which aspects of cognition in general are computable;

computational psychology studies *human* cognition using computational techniques [cf. Shapiro 1992; Rapaport 2003].) This is a distinction that Searle does not seem to make. Finally, Pylyshyn's version of "cognitivism" (1985) does not claim that the brain *is* a digital computer, but that mental processes are computational processes. That seems to me to be compatible with the brain being "more" than a digital computer.

Searle complains that multiple realizability is "disastrous" (p. 26; cf. Rapaport 2005, §4.1). The first reason is that *anything* can be described in terms of 0s and 1s (p. 26), and there might be *lots* of 0-1 encodings of the brain. But the real question, it seems to me, is this: Does the brain *compute* (effectively) some function? What is the input-output description of that function? The answer to the latter question is whatever psychology tells us is intelligence, cognition, etc. For special cases, it's easier to be a bit more specific: for natural-language understanding, the input is some utterance of natural language, and the output is an "appropriate" response (where the measure of "appropriateness" is defined, let's say, sociolinguistically). For vision, the input is some physical object, and the output is, again, some "appropriate" response (say, an utterance identifying the object or some scene, or some behavior to pick up or avoid the object, etc.). Moreover, these two modules (natural-language understanding and vision) must be able to "communicate" with each other. (They might or might not be modular in Fodor's sense [1983], or cognitively impenetrable in Pylyshyn's sense [1985]. In any case, solving one of these problems will require a solution to the other; they are "AI-complete" [Shapiro 1992].)

The second allegedly disastrous consequence of multiple realizability is that "syntax is not intrinsic to physics. The ascription of syntactical properties is always relative to an agent or observer who treats certain physical phenomena as syntactical" (p. 26). The observer assigns 0s and 1s to the physical phenomena. But Morris's definition of syntax as relations among symbols (uninterpreted marks) can be extended to relations among components of any system. Surely, physical objects stand in those relationships "intrinsically." And if 0s and 1s *can* be ascribed to a physical object (by an observer), that *fact* exists independently of the agent who *discovers* it.

Searle's claim "that syntax is essentially an observer relative notion" (p. 27) is very odd. One would have expected him to say that about *semantics*, not syntax. Insofar as one can look at a complex system and describe (or discover) relations among its parts (independently of any claims about what it does at any higher level), one is doing *non*-observer-relative syntax. Searle says that "this move is no help. A physical state of a system is a computational state only relative to the assignment to that state of some computational role, function, or interpretation" (p. 27), where, presumably, the assignment is made by an observer. But an assignment is an assignment of meaning; it's an interpretation. So is Searle saying that computation is fundamentally a *semantic* notion? But, for Church, Turing, et al., computation is purely *syntactic*. It's only the input-output coding that *might* constitute an assignment. But such coding is only needed in order to be able to link the syntax with the standard theory of computation in terms of functions from natural numbers to natural numbers. If we're willing to express the theory of computation in terms of functions from physical states to physical states (and why shouldn't we?), then it's not relative.

Searle rejects question 1: "There is no way you could discover that something is intrinsically a digital computer because the characterization of it as a digital computer is always relative to an observer who assigns a *syntactical* interpretation to the purely physical features of the system" (p. 28, my italics). I, too, reject question 1, but for a very different reason: I think the question is really whether *mental processes* are computational. In any event, suppose we *do* find computer programs that exhibit intelligent input-output behavior, i.e., that pass the Turing Test. Computational *philosophy* makes no claim about whether that tells us that the human *brain* is a digital computer. It only tells us that intelligence is a computable function. So, at best, Searle's arguments are against computational *psychology*. But even that need not imply that the brain *is* a digital computer, only that it behaves as if it were. To discover that something *X* is intrinsically a digital computer, or a *Y*, is to have an abstraction *Y*, and to find correspondences between *X* and *Y*.

Perhaps what Searle is saying is that being computational is not a *natural* kind but an artifactual kind (cf. Churchland and Sejnowski 1992):

> I am not saying there are *a priori* limits on the patterns we could discover in nature. We could no doubt discover a pattern of events in my brain that was isomorphic to the implementation of the vi program on this computer. (Searle 1990, 28)

This is to admit what I observed two paragraphs back. Searle continues:

> But to say that something is *functioning as* a computational process is to say something more than that a pattern of physical events is occurring. It requires the assignment of a computational interpretation by some agent. (Searle 1990, 28)

But why? Possibly because to find correspondences between two things (say, a brain and the Abstraction Computational Process—better, the Abstraction Computer) is observer-relative? But if we have *already* established that a certain brain process is an implementation of vi, what *extra* "assignment of a computational interpretation by some agent" is needed?

Searle persists:

> Analogously, we might discover in nature objects which had the same sort of shape as chairs and which could therefore be used as chairs; but we could not discover objects in nature which were functioning as chairs, except relative to some agent who regarded them or used them as chairs. (Searle 1990, 2)

The analogy is clearly with *artifacts*. But the notion of a computational process does not seem to me to be artifactual; it is *mathematical*. So the proper analogy would be something like this: Can we discover in nature objects that were, say, sets, or numbers, or Abelian groups? Here, the answer is, I think, (a qualified) "yes." (It is qualified because sets and numbers are abstract and infinite, while the world is concrete and finite. Groups may be a clearer case.) In any event, is Searle claiming that the implementation of vi in my brain isn't vi until someone *uses it as* vi? If there is an implementation of vi on my Macintosh that no one ever uses, it's still vi.

Searle accuses computational cognitive scientists of "commit[ing] the homunculus fallacy...treat[ing] the brain as if there were some agent inside it using it to compute with" (p. 28). But consider Patrick Hayes's (1990; 1997) objection to the Chinese-Room Argument: computation is a series of switch-settings; it isn't rule-following. (On this view, by the way, the solar system *does* compute certain mathematical functions.) Turing machines do *not* follow rules; they simply change state. There are, however, descriptions—programs—of the state changes, and anything that follows (executes) that program computes the same function computed by the Turing machine. A *universal* Turing machine can also follow that program. But the original, special-purpose Turing machine's program is "hardwired" (an analogy, of course, since everything is abstract here). A universal Turing machine has *its* program similarly hardwired. It is only when the universal Turing machine is fed a program that it follows the rules of that program. But that's what *we* do when *we* consciously follow (hand-simulate) the rules of a program. So it's *Searle* who commits the homuncular fallacy in the Chinese-Room Argument by putting a person in the room. It is not the person in the room who either does or does not understand Chinese; it is the entire system (Rapaport 2000; 2006). Similarly, it is not some *part* of my brain that understands language; it is *I* who understands.

In his discussion of "discharging" the homunculus, Searle says that "all of the higher levels reduce to this bottom level. Only the bottom level really exists; the top levels are all just *as-if*" (p. 29). But *all* levels exist, and *all* levels "do the same thing," albeit in different ways (Rapaport 1990; 2005).

I noted above that systems that don't follow rules can still be said to be computing. My example was the solar system. Searle offers "nails [that] compute the distance they are to travel in the board from the impact of the hammer and the density of the wood" (p. 29) and the human visual system; "neither," according to him, "compute anything" (p. 29). But, in fact, they both do. (The nail example might not be ideal, but it's a nice example of an *analog* computation.)

> But you do not *understand* hammering by supposing that nails are somehow intrinsically implementing hammering algorithms and you do not *understand* vision by supposing the system is implementing, e.g., the shape from shading algorithm. (Searle 1990, 29; my italics.)

Why not? It gives us a theory about how the system might be performing the task. We can falsify (or test) the theory. What more could *any* (scientific) theory give us? What further kind of understanding could there be? Well, there could be first-person understanding, but I doubt that we could ever know what it is like to be a nail or a solar system. We *do* understand what it is like to be a cognitive agent!

The problem, I think, is that Searle and I are interested in different (but complementary) things:

> ...you cannot explain a physical system such as a typewriter or a brain by identifying a pattern which it shares with its computational simulation, because the existence of the pattern does not explain how the system actually works *as a physical system*. (Searle 1990, 32)

Of course not. That would be to confuse the implementation with the Abstraction. Searle is interested in the former; he wants to know *how the (human) brain works*. I, however, want to know *what the brain does* and how *anything* could do it. For that, I need an account at the functional/computational level, not a biological (or neuroscientific) theory.

> The mistake is to suppose that in the sense in which computers are used to process information, brains also process information. [Cf. Johnson 1990.] To see that that is a mistake, contrast what goes on in the computer with what goes on in the brain. In the case of the computer, an outside agent encodes some information in a form that can be processed by the circuitry of the computer. That is, he or she provides a syntactical realization of the information that the computer can implement in, for example, different voltage levels. The computer then goes through a series of electrical stages that the outside agent can interpret both syntactically and semantically even though, of course, the hardware has no intrinsic syntax or semantics: It is all in the eye of the beholder. And the physics does not matter provided only that you can get it to implement the algorithm. Finally, an output is produced in the form of physical phenomena which an observer can interpret as symbols with a syntax and a semantics.
>
> But now contrast this with the brain...none of the relevant neurobiological processes are observer relative...and the specificity of the neurophysiology matters desperately. (Searle 1990, 34)

There is much to disagree with here. First, "an outside agent" need *not* "encode...information in a form that can be processed by the circuitry of the computer." A computer could be (and typically is) designed to take input directly from the real world and to perform the encoding (better: the transduction) itself, as, e.g., in document-image understanding (cf. Srihari and Rapaport 1989; 1990; Srihari 1991; 1993; 1994). Conversely, abstract concepts are "encoded" in natural language so as to be processable by *human* "circuitry."

Second, although I find the phrase "syntactical realization" quite congenial (cf. Rapaport 1995), I'm not sure how to parse the rest of the sentence in which it appears. What does the computer "implement in voltage levels": the information? the syntactical realization? I'd say the former, and that the syntactical realization *is* the voltage levels. So there's an issue here of whether the voltage levels are *interpreted as* information, or vice versa.

Third, the output need not be physical phenomena interpreted by an observer as symbols. The output *could* be an action, or more internal data (e.g., as in a vision system),[2] or even natural language to be interpreted by another *computer*. Indeed, the latter suggests an interesting research project: set up Cassie and Oscar, two computational cognitive agents implemented in a knowledge-representation, reasoning, and acting system such as SNePS.[3] Let Cassie have a story pre-stored or as the result of "reading" or "conversing." Then let her tell the story to Oscar and ask him questions about it. No *humans* need be involved.

Fourth, neurobiological processes aren't observer-relative, only because we don't care to, or need to, describe them that way. The computer works as it does independently of us, too. Of course, for *us* to understand what the brain is doing—from a third-person point of view—we need a psychological level of description (cf. Chomsky 1968; Fodor 1968; Enç 1982).

Finally, why should "the specificity of the neurophysiology matter desperately"? Does this mean that if the neurophysiology were different, it wouldn't be a human brain? I suppose so, but that's relevant only for the implementation side of the issue, not the Abstraction side, with which I am concerned.

Here is another example of how Searle does not seem to understand what computational cognitive science is about:

> A standard computational model of vision will take in information about the visual array on my retina and eventually print out the sentence, "There is a car coming toward me." But that is not what happens in the actual biology. In the biology a concrete and specific series of electro-chemical reactions are set up by the assault of the photons on the photo receptor cells of my retina, and this entire process eventually results in a concrete visual experience. The biological reality is not that of a bunch of words or symbols being

> produced by the visual system, rather it is a matter of a concrete specific conscious visual event; this very visual experience. (Searle 1990, 34-35)

The first sentence is astounding. First, why does he assume that the input to the computational vision system is *information on the retina*, rather than *things in the world*? The former is close to an *internal* symbol representing external information! Second, it is hardly "standard" to have a vision system yield a *sentence* as an output. It might, of course ("Oh, what a pretty red flower."), but, in the case of a car coming at the system, an aversive maneuver would seem to be called for, not a matter-of-fact description. Nonetheless, precisely that input-output interaction *could*, *pace* Searle, be "what happens in the actual biology": I could say that sentence upon appropriate retinal stimulation.

Of course, as the rest of the quotation makes clear, Searle is more concerned with the intervening qualitative experience, which, he seems to think, humans have but computers don't (or can't). Well, could they? Surely, there ought to be an intervening stage in which the retinal image is processed (perhaps stored) before the information thus processed or stored is passed to the natural-language module and interpreted and generated. Does that process have a qualitative feel? Who knows? *How* would you know? Indeed, how do I know (or believe) that *you* have such a qualitative feel? The question is the same for both human and computer. Stuart C. Shapiro has suggested how a pain-feeling computer could be built (Rapaport 2005, §2.3.1); similarly, it's possible that a physical theory of sensation could be constructed. Would it be computational? Perhaps not—but so what? Perhaps some "mental" phenomena are not *really* mental (or computational) after all (Rapaport 2005, §2.3). Or perhaps a computational theory will always be such that there is a role to play for some sensation or other, even though the actual sensation in the event is not computational. That is, every computational theory of pain or vision or what have you will be such that it will refer to a sensation without specifying what the sensation is. (Cf. Gracia's [1990] example of a non-written universal for a written text, discussed in Rapaport 2005, §2.2. See also McDermott 2001.)

Of course, despite my comments about the linguistic output of a vision system, the sentence that Searle talks about could be a "sentence" of one's language of thought. That, however, would fall under the category of being a "concrete specific conscious visual event" and "not...a bunch of words or symbols" (Cf. Pylyshyn 1981; Srihari op. cit.; Srihari & Rapaport op. cit.).

Searle's final point about question 1 is this:

> The point is not that the claim "The brain is a digital computer" is false. Rather it does not get up to the level of falsehood. It does not have a clear sense. (Searle 1990, 35)

This is because "you could not *discover* that the brain *or anything else* was intrinsically a digital computer" (p. 35, my italics). "Or anything else"? Even an IBM PC? Surely not. Possibly he means something like this: suppose we find an alien physical object and theorize that it is a digital computer. Have we *discovered* that it is? No—we've got an *interpretation* of it *as* a digital computer (cf. "you could assign a computational interpretation to it as you could to anything else" [p. 35]). But how else *could* we "discover" anything about it? Surely, we could discover that it's made of silicon and has $10^k$ parts. But that's consistent with his views about *artifacts*. Could we *discover* the topological arrangement of its parts? I'd say, "yes." Can we *discover* the sequential arrangement of its behaviors? Again, I'd say, "yes." Now consider this: How do we determine that it's made of silicon? By subjecting it to certain physical or chemical tests and having a theory that says that any substance that *behaves* thus and so is (made of) silicon. But if anything that behaves such and thus is a computer, then so is this machine! So we *can* discover that (or whether) it is a computer. (Better: we can discover whether its processing is computational.)

**Endnotes**

1. Angier (1992) reports that "sperm cells possess the same sort of odor receptors that allow the nose to smell." This does not mean, of course, that sperm cells have the mental capacity to have smell-qualia. Blakeslee (1993) reports that "humans… may exude…odorless chemicals called pheromones that send meaningful signals to other humans." She calls this "a cryptic sensory system that exists without conscious awareness…." And Fountain (2006) discusses a plant that has what might be called a sense of smell, presumably without any smell-qualia.
2. Searle seems to think (p. 34) that vision systems yield sentences as output! (See below.)
3. Shapiro 1979; 2000; Shapiro and Rapaport 1987; 1992; 1995; Shapiro et al. 2006. Further information is available online at: http://www.cse.buffalo.edu/sneps and at: http://www.cse.buffalo.edu/~rapaport/snepskrra.html.

**References**

Angier, Natalie. "Odor Receptors Discovered in Sperm Cells." *The New York Times* (January 30): A19, 1992.

Blakeslee, Sandra. "Human Nose May Hold An Additional Organ For a Real Sixth Sense." *The New York Times* (September 7): C3, 1993.

Chomsky, Noam. 1968. *Language and Mind*. New York: Harcourt, Brace & World.

Churchland, Patricia S., and Terrence J. Sejnowski. 1992. *The Computational Brain*. Cambridge, MA: MIT Press.

Crane, Tim. 1998. "Intentionality as the Mark of the Mental." *Current Issues in the Philosophy of Mind*, edited by Anthony O'Hear. Cambridge, UK: Cambridge University Press.

Enç, Berent. "Intentional States of Mechanical Devices." *Mind* 91 (362; April 1982): 161-82.

Fetzer, James H. "Program Verification: The Very Idea." *Communications of the ACM* 31 (1988): 1048-63.

Fetzer, James H. "Philosophical Aspects of Program Verification." *Minds and Machines* 1 (1991): 197-216.

Fodor, Jerry A. 1968. *Psychological Explanation: An Introduction to the Philosophy of Psychology*. New York: Random House.

Fodor, Jerry A. 1983. *The Modularity of Mind: An Essay in Faculty Psychology*. Cambridge, MA: MIT Press.

Fountain, Henry. "This Plant Has the Sense of Smell (Loves Tomatoes, Hates Wheat)." *The New York Times* (October 3): F1, 2006.

Gracia, Jorge J.E. "Texts and Their Interpretation." *Review of Metaphysics* 43 (1990): 495-542.

Haugeland, John. 1985. *Artificial Intelligence: The Very Idea*. Cambridge, MA: MIT Press.

Hayes, Patrick J. 1990. "Searle's Chinese Room." Abstract of presentation to the Society for Philosophy and Psychology Symposium on Searle's Chinese Room and Workshop on Symbol Grounding. University of Maryland (electronic mail posting from Stevan Harnad, June 6, 1990).

Hayes, Patrick J. "What Is a Computer? An Electronic Discussion." *The Monist* 80:3 (1997).

Johnson-Laird, Philip N. 1988. *The Computer and the Mind: An Introduction to Cognitive Science*. Cambridge, MA: Harvard University Press.

McDermott, Drew. 2001. *Mind and Mechanism*. Cambridge, MA: MIT Press.

Morris, Charles. 1938. *Foundations of the Theory of Signs*. Chicago, IL: University of Chicago Press.

Pylyshyn, Zenon. 1981. "The Imagery Debate: Analog Media versus Tacit Knowledge," edited by Ned Block. In *Imagery*. Cambridge, MA: MIT Press.

Pylyshyn, Zenon. 1985. *Computation and Cognition: Toward a Foundation for Cognitive Science*, 2nd ed. Cambridge, MA: MIT Press.

Rapaport, William J. "Meinongian Theories and a Russellian Paradox." *Noûs* 12: (1978): 153-80; errata, *Noûs* 13 (1979): 125.

Rapaport, William J. 1990. "Computer Processes and Virtual Persons: Comments on Cole's 'Artificial Intelligence and Personal Identity'." *Technical Report*. Buffalo: SUNY Buffalo Department of Computer Science. http://www.cse.buffalo.edu/~rapaport/Papers/cole.tr.17my90.pdf.

Rapaport, William J. 1995. "Understanding Understanding: Syntactic Semantics and Computational Cognition," edited by James E. Tomberlin. In *Philosophical Perspectives, Vol. 9: AI, Connectionism, and Philosophical Psychology*. Atascadero, CA: Ridgeview; reprinted in *Language and Meaning in Cognitive Science: Cognitive Issues and Semantic Theory, Artificial Intelligence and Cognitive Science: Conceptual Issues,* Vol. 4, edited by Toribio, Josefa, & Clark, Andy. 1998. New York: Garland.

Rapaport, William J. 1996. "Understanding Understanding: Semantics, Computation, and Cognition." *Technical Report*. Buffalo: SUNY Buffalo Department of Computer Science.

Rapaport, William J. "How Minds Can Be Computational Systems," *Journal of Experimental and Theoretical Artificial Intelligence* 10 (1998): 403-19.

Rapaport, William J. "Implementation is Semantic Interpretation," *The Monist* 82 (1999): 109-30.

Rapaport, William J. "How to Pass a Turing Test: Syntactic Semantics, Natural-Language Understanding, and First-Person Cognition," *Journal of Logic, Language, and Information* 9 (2000): 467-90; reprinted in *The Turing Test: The Elusive Standard of Artificial Intelligence*, edited by James H. Moor. 2003. Dordrecht: Kluwer.

Rapaport, William J. "Holism, Conceptual-Role Semantics, and Syntactic Semantics," *Minds and Machines* 12 (2002): 3-59.

Rapaport, William J. "What Did You Mean by That? Misunderstanding, Negotiation, and Syntactic Semantics," *Minds and Machines* 13 (2003): 397-427.

Rapaport, William J. "Implementation Is Semantic Interpretation: Further Thoughts," *Journal of Experimental and Theoretical Artificial Intelligence* 17 (2005): 385-417.

Rapaport, William J. 2006. "How Helen Keller Used Syntactic Semantics to Escape from a Chinese Room." http://www.cse.buffalo.edu/~rapaport/Papers/helenkeller.pdf.

Searle, John R. "Is the Brain a Digital Computer?" *Proceedings and Addresses of The American Philosophical Association* 64 (1990): 21-37.

Shapiro, Stuart C. 1979. "The SNePS Semantic Network Processing System." In *Associative Networks: Representation and Use of Knowledge by Computers*, edited by Nicholas Findler. New York: Academic Press.

Shapiro, Stuart C. 1992. "Artificial Intelligence." In *Encyclopedia of Artificial Intelligence*, 2nd edition, edited by Stuart C. Shapiro. New York: John Wiley & Sons.

Shapiro, Stuart C. 2000. "SNePS: A Logic for Natural Language Understanding and Commonsense Reasoning." In *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, edited by [L]ucja M. Iwanska & Stuart C. Shapiro. Menlo Park, CA/Cambridge, MA: AAAI Press/MIT Press.

Shapiro, Stuart C., and William J. Rapaport. 1987. "SNePS Considered as a Fully Intensional Propositional Semantic Network." In *The Knowledge Frontier: Essays in the Representation of Knowledge*, edited by Nick Cercone & Gordon McCalla. New York: Springer-Verlag; shorter version appeared in *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI-86, Philadelphia)*. Los Altos, CA: Morgan Kaufmann; a revised shorter version appears as "A Fully Intensional Propositional Semantic Network." In *Philosophy and the Computer*, edited by Leslie Burkholder. 1992. Boulder, CO: Westview Press.

Shapiro, Stuart C., and William J. Rapaport. "The SNePS Family." *Computers and Mathematics with Applications* 23 (1992): 243-75; reprinted in *Semantic Networks in Artificial Intelligence*, edited by F. Lehmann. 1992. Oxford: Pergamon Press.

Shapiro, Stuart C., and William J. Rapaport. 1995. "An Introduction to a Computational Reader of Narratives." In *Deixis in Narrative: A Cognitive Science Perspective*, edited by Judith F. Duchan, Gail A. Bruder, and Lynne E. Hewitt. Hillsdale, NJ: Lawrence Erlbaum Associates.

Shapiro, Stuart C., and the SNePS Research Group. 2006. "SNePS." http://en.wikipedia.org/wiki/SNePS.

Smith, Brian Cantwell. 1985. "Limits of Correctness in Computers." Reprinted in *Computerization and Controversy*, edited by Charles Dunlop & Rob Kling. 1991. San Diego: Academic Press.

Srihari, Rohini K. 1991. "PICTION: A System that Uses Captions to Label Human Faces in Newspaper Photographs." *Proceedings of the 9th National Conference on Artificial Intelligence (AAAI-91, Anaheim)*. Menlo Park, CA: AAAI Press/MIT Press.

Srihari, Rohini K. 1993. "Intelligent Document Understanding: Understanding Photos with Captions." *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR-93, Tsukuba City, Japan)*.

Srihari, Rohini K. 1994. "Use of Collateral Text in Understanding Photos in Documents." *Proceedings of the Conference on Applied Imagery and Pattern Recognition (AIPR/SPIE, Washington, DC)*.

Srihari, Rohini K., and William J. Rapaport. 1989. "Extracting Visual Information From Text: Using Captions to Label Human Faces in Newspaper Photographs." *Proceedings of the 11th Annual Conference of the Cognitive Science Society (Ann Arbor, MI)*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Srihari, Rohini K., and William J. Rapaport. 1990. "Combining Linguistic and Pictorial Information: Using Captions to Interpret Newspaper Photographs." In *Current Trends in SNePS—Semantic Network Processing System*, edited by Deepak Kumar. Berlin: Springer-Verlag Lecture Notes in Artificial Intelligence, 437.

Tenenbaum, Aaron M., and Moshe J. Augenstein. 1981. *Data Structures Using Pascal*. Englewood Cliffs, NJ: Prentice-Hall.

Turing, Alan M. "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society* 42 (1936): 230-65; reprinted, with corrections, in *The Undecidable: Basic Papers On Undecidable Propositions, Unsolvable Problems and Computable Functions*, edited by Martin Davis. 1965. New York: Raven Press.

Twardowski, Kasimir. 1977, 1894. *On the Content and Object of Presentations, Reinhardt Grossmann* (trans.). The Hague: Nijhoff.

Wartofsky, Marx W. 1966, 1979. "The Model Muddle: Proposals for an Immodest Realism." In *Models: Representation and the Scientific Understanding*. Dordrecht, Holland: 1-11. D. Reidel.

Wartofsky, Marx W. 1979. "Introduction." In *Models: Representation and the Scientific Understanding*. Dordrecht, Holland: xiii-xxvi. D. Reidel.

---

# *Epistemology Naturalized: The Info-Computationalist Approach*

**Gordana Dodig-Crnkovic**
*Mälardalen University, Västerås, Sweden*

Naturalized epistemology (Feldman, Kornblith, Stich) is, in general, an idea that knowledge may be studied as a natural phenomenon—that the subject matter of epistemology is not our concept of knowledge, but the knowledge itself.

In his "Epistemology Naturalized," Quine claims the following:

> The stimulation of his sensory receptors is all the evidence anybody has had to go on, ultimately, in arriving at his picture of the world. Why not just see how this construction really proceeds? Why not settle for psychology? (Quine 1969)

This essay will re-phrase the question to be: Why not settle for computing? The main reason is that info-computationalism provides a unifying framework that makes it possible for different research fields such as philosophy, computer science, neuroscience, cognitive science, biology, and a number of others to communicate.

We will give an account of the naturalized epistemology based on the computational character of cognition and agency—which includes evolutionary approaches to cognition (Lorenz 1977, Popper 1978, Toulmin 1972 and Campbell et al. 1989, Harms 2004). In this framework knowledge is seen as a result of the structuring of input data (data Y information Y knowledge) by an interactive computational process going on in the nervous system during the adaptive interplay of an agent with the environment, which clearly increases its ability to cope with the dynamics of the world.

Traditionally, there is a widely debated problem of representation of information and the role of representation in explaining and producing information, a discussion about two seemingly incompatible views: a hard, explicit, and static notion of representation versus an implicit and dynamic (interactive) one. The central point is that both of those views are eminently info-computational. Within an info-computational framework, those classical (Turing-machine type) and connectionist views are reconciled and used to describe different aspects of cognition (Arnellos et al. 2005, Dawson 2006). The project of naturalizing epistemology through info-computationalism builds on the development of multilevel dynamical computational models and simulations of a nervous system and has important consequences for the development of intelligent systems and artificial life.

## 1. Dual Aspect Info-Computational Framework

Within the field of computing and philosophy, two distinct branches have been established: informationalism (in which the focus is on information as the stuff of the universe; Floridi 2002) and computationalism (where the universe is seen as a computer). Chaitin (2006) mentions cellular automata[1] researchers—and computer scientists Fredkin, Wolfram, Toffoli, and Margolus, and physicists Wheeler, Zeilinger, 't Hooft, Smolin, Lloyd, Zizzi, Mäkelä, and Jacobson, as prominent computationalists.

Recently, a synthetic approach has been proposed in the form of dual-aspect info-computationalism (Dodig-Crnkovic 2006), in which the universe is viewed as a structure (information) in a permanent process of change (computation). According to this view, information and computation constitute two aspects of reality and, like particle and wave, or matter and energy, capture different facets of the same physical world.

Computation may be either discrete or continuous (digital or analogue). The present approach offers a generalization of traditional computationalism in the sense that "computation" is understood as the process governing dynamics of the universe, or in the words of Chaitin:

> And how about the entire universe, can it be considered to be a computer? Yes, it certainly can, it is constantly computing its future state from its current state, it's constantly computing its own time-evolution! And as I believe Tom Toffoli pointed out, actual computers like your PC just hitch a ride on this universal computation! (Chaitin 2006)

Mind is seen in this dual-aspect framework as a computational process on an informational structure that, both in its digital and analogue forms, occurs through changes in the structures of our brains and bodies as a consequence of interaction with the physical universe. This approach leads to a naturalized, evolutionary epistemology that understands cognition as a phenomenon that can be ascribed, in the spirit of Maturana and Varela, even to simplest living organisms, and in the same vein to artificial life.

In order to be able to comprehend and eventually construct artificial cognitive systems we can learn from the historical development of biological cognitive functions and structures from the simple ones upward. A very interesting account of developmental ascendancy, from bottom-up to top-down control, is given by Coffman (2006).

### 1.1 Natural Computation Beyond the Turing Limit

As a direct consequence of the computationalist view that every natural process is computation in a computing universe, "computation" must be generalized to mean *natural* computation. MacLennan (2004) defines "natural computation" as "computation occurring in nature or inspired by that in nature," which includes quantum computing and molecular computation, and may be represented by either discrete or continuous models. Examples of computation occurring in nature include information processing in evolution by natural selection, in the brain, in the immune system, in the self-organized collective behavior of groups of animals such as ant colonies, and in particle swarms. Computation inspired by nature includes genetic algorithms, artificial neural nets, simulated immune systems, ant colony optimization, particle swarm optimization, and so forth. There is a considerable synergy gain in relating human-designed computing with the computing in nature. Chaitin claims that "we only understand something if we can program it." In the iterative course of modeling and computationally simulating (programming) natural processes, we learn to reproduce and predict more and more of the characteristic features of the natural systems.

Ideal, classical, theoretical computers are mathematical objects and are equivalent to algorithms, abstract automata (Turing machines), effective procedures, recursive functions, or formal languages. Compared with new emerging computing paradigms, particularly interactive computing and natural computing, Turing machines form the proper subset of the set of information processing devices.

An interesting new situation (Wegner 1998) arises when the computer is conceived as an open system in communication with the environment, the boundary of which is dynamic, as in biological systems. Chaisson (2002), for example, defines life as an "open, coherent, space-time structure maintained far from thermodynamic equilibrium by a flow of energy through it." On a computationalist view, organisms are constituted by computational processes; they are "living computers." In the living cell an info-computational process takes place using DNA in an open system exchanging information, matter, and energy with the environment.

Burgin (2005) identifies three distinct components of information processing systems: hardware (physical devices), software (programs that regulate its functioning and sometimes can be identical with hardware, as in biological computing), and infoware (information processed by the system). Infoware is a shell built around the software-hardware core, which is the traditional domain of automata and algorithm theory. Semantic Web is an example of infoware that is adding semantic component to the information present on the web (Berners-Lee 2001, Hendler and Lassila 2001).

For implementations of computationalism, interactive computing is the most appropriate general model, as it naturally suits the purpose of modeling a network of mutually communicating processes (Dodig-Crnkovic 2006). It will be of particular interest to computational accounts of epistemology, as a cognizing agent interacts with the environment in order to gain experience and knowledge. It also provides the natural unifying framework for reconciliation of classical and connectionist views of cognition.

## 2. Epistemology Naturalized

> Indeed, cognitive ethologists find the only way to make sense of the cognitive equipment in animals is to treat it as an information processing system, including equipment for perception, as well as the storage and integration of information; that is, after all, the point of calling it cognitive equipment. That equipment which can play such a role confers selective advantage over animals lacking such equipment no longer requires any argument. (Kornblith 1999)

Our specific interest is in how the structuring from data to information and knowledge develops on a phenomenological level in a cognitive agent (biological or artificial) in its interaction with the environment. The central role of interaction is expressed by Goerzel (1994):

> Today, more and more biologists are waking up to the sensitive environment-dependence of fitness, to the fact that the properties which make an organism fit may not even be present in the organism, but may be emergent between the organism and its environment.[2]

One can say that living organisms are "about" the environment, that they have developed adaptive strategies to survive by internalizing environmental constraints. The interaction between an organism and its environment is realized through the exchange of physical signals that might be seen as data or, when structured, as information. Organizing and mutually relating different pieces of information results in knowledge. In that context, computationalism appears as the most suitable framework for naturalizing epistemology.

A very interesting idea presented by Maturana and Varela (1980) is that even the simplest organisms possess cognition and that their meaning-production apparatus is contained in their metabolism. Of course, there are also non-metabolic interactions with the environment, such as locomotion, that also generate meaning for an organism by changing its environment and providing new input data. We will take Maturana's and Varelas' theory as the basis for a computationalist account of the evolutionary epistemology.

At the physical level, living beings are open, complex, computational systems in a regime on the edge of chaos, characterized by maximal informational content. Complexity is found between orderly systems with high information compressibility and low information content and random systems with low compressibility and high information content.

> Langton has compared these different regions to the different states of matter. Fixed points are like crystals in that they are for the most part static and orderly. Chaotic dynamics are similar to gases, which can be described only statistically. Periodic behavior is similar to a non-crystal solid, and complexity is like a liquid that is close to both the solid and the gaseous states. In this way, we can once again view complexity and computation as existing on the edge of chaos and simplicity. (Flake 1998)

Artificial agents may be treated analogously with animals in terms of different degrees of complexity; they may range from software agents with no sensory inputs at all to cognitive robots with varying degrees of sophistication of sensors and varying bodily architecture.

> The question is: how does information acquire meaning naturally in the process of an organism's interaction with its environment? A straightforward approach to naturalized epistemology attempts to answer this question via study of evolution and its impact on the cognitive, linguistic, and social structures of living beings, from the simplest ones to those at highest levels of organizational complexity.[3] (Bates 2005)

Animals are equipped with varying physical hardware, sets of sensory apparatuses (compare an amoeba with a mammal), and goals and behaviors. For different animals, the "aboutness" concerning the same physical reality is different in terms of causes and their effects.

Thus, the problematic aspect of any correspondence theory (including spectator models of representation) is the difficulty of deciding whose reality is to be considered "the true one." However, Harms (2004) claims that "we now have a fairly satisfactory account of correspondence truth for simple signals like animal warning cries, a rather surprising triumph for naturalism. Essentially, a signal in an environmental tracking system is true when it gets its timing right vis-à-vis its adaptive design" (also see Millikan 1984, Skyrms 1996). The correspondence is in this case about the existence of the phenomenon ("there is a cat") and not about the "true nature of the phenomenon" (its interpretation).

An agent receives inputs from the physical environment (data) and interprets these in terms of its own earlier experiences, comparing them with stored data in a feedback loop.[4] Through that interaction between the environmental data and the inner structure of an agent, a dynamical state is obtained in which the agent has established a representation of the situation. The next step in the loop is to compare the present state with its goals and preferences (saved in an associative memory). This process results in the anticipation of what various actions from the given state might have for consequences (Goertzel 1994). Here is an alternative formulation:

> This approach is not a hybrid dynamic/symbolic one, but interplay between analogue and digital information spaces, in an attempt to model the representational behavior of a system. The focus on the explicitly referential covariation of information between system and environment is shifted towards the interactive modulation of implicit internal content and therefore, the resulting pragmatic adaptation of the system via its interaction with the environment. The basic components of the framework, its nodal points and their dynamic relations are analyzed, aiming at providing a functional framework for the complex realm of autonomous information systems. (Arnellos et al. 2005)

### 2.1 Interactive Naturalism and Computational Process

Interactivism[5] (Birkhard 2004, Kulakov & Stojanov 2002) is a philosophical approach especially suited to the analysis of agency. On the ontological level, it involves (emergentist) naturalism, which means that the physical world (matter) and mind are integrated, mind being an emergent property of a physical process. It is closely related to process metaphysics (Whitehead 1978), in which the fundamental nature of the universe is understood as organization of processes.

Interactivism has been applied to a range of phenomena, including perception, consciousness, learning, language, memory, emotions, development, personality, rationality, biological functionality, and evolution. The approach is inspired by, among others, Piaget's interactionism and constructivism (Piaget 1987), but it differs from Piaget in that it gives a central role to variational construction and selection.

The interactive model is pragmatist in its process and action approach, and in its focus on the consequences of interaction it resembles Peirce's model of meaning. The essential difference between the interactivist concept of perception and Peirce's concept is the emphasis in the former on the process (interactive) nature of perception (data) and information (representation).

## 2.2 Evolutionary Development

> One cannot account for the functional architecture, reliability, and goals of a nervous system without understanding its adaptive history. Consequently, a successful science of knowledge must include standard techniques for modeling the interaction between evolution and learning. (Harms 2005)

A central question is thus what the mechanism is of the evolutionary development of cognitive abilities in organisms. Critics of the evolutionary approach mention the impossibility of "blind chance" to produce such highly complex structures as intelligent living organisms. Proverbial monkeys typing Shakespeare are often used as an illustration (an interesting account is given by Gell-Man in his *Quark and the Jaguar*). However, Lloyd (2006) mentions a very good counterargument, originally due to Chaitin and Bennet. The "typing monkeys" argument does not take into account physical laws of the universe, which dramatically limit what can be typed. Moreover, the universe is not a typewriter, but a computer, so a monkey types random input into a computer. The computer interprets the strings as programs.

> Quantum mechanics supplies the universe with "monkeys" in the form of random fluctuations, such as those that seeded the locations of galaxies. The computer into which they type is the universe itself. From a simple initial state, obeying simple physical laws, the universe has systematically processed and amplified the bits of information embodied in those quantum fluctuations. The result of this information processing is the diverse, information-packed universe we see around us: programmed by quanta, physics give rise first to chemistry and then to life; programmed by mutation and recombination, life gave rise to Shakespeare; programmed by experience and imagination, Shakespeare gave rise to Hamlet. You might say that the difference between a monkey at a typewriter and a monkey at a computer is all the difference in the world. (Lloyd 2006)

Allow me to add one comment on Lloyd's computationalist claim. The universe/computer on which a monkey types is at the same time the hardware and the program, in a way similar to the Turing machine. (An example from biological computing is the DNA where the hardware [the molecule] is at the same time the software [the program, the code]). In general, each new input restructures the computational universe and changes the preconditions for future inputs. Those processes are interactive and self-organizing. That makes the essential speed-up for the process of getting more and more complex structures.

## 2.3 Info-Computational Complexity of Cognition

> Dynamics lead to statics, statics leads to dynamics, and the simultaneous analysis of the two provides the beginning of an understanding of that mysterious process called mind. (Goertzel 1994)

In the info-computationalist vocabulary, "statics" (structure) corresponds to "information" and "dynamics" corresponds to "computation."

One question that now may be asked is: Why doesn't an organism exclusively react to data as received from the world/environment? Why is information used as building blocks, and why is knowledge constructed? In principle, one could imagine a reactive agent that responds directly to input data without building an informational structure out of raw input.

The reason may be found in the computational efficiency of the computation concerned. Storage of data that are constant or are often reused saves enormous amounts of time. So, for instance, if instead of dealing with each individual pixel in a picture we can make use of symbols or patterns that can be identified with similar memorized symbols or patterns, the picture can be handled much more quickly.

Studies of vision show that cognition focuses on that part of the scene which is variable and dynamic and uses memorized data for the rest that is static (this is the notorious frame problem of AI). Based on the same mechanism, we use ideas already existing to recognize, classify, and characterize phenomena. Our cognition is thus an emergent phenomenon, resulting from both memorized (static) and observed (dynamic) streams. Forming chunks of structured data into building blocks (instead of performing time-consuming computations on those data sets in real time) is an enormously powerful acceleration mechanism. With each higher level of organization, the computing capacity of an organism's cognitive apparatus is further increased. The efficiency of meta-levels is becoming evident in computational implementations.

Cognition as the multilevel control network in Goertzel's model is "pyramidal" in the sense that each process is connected to more processes below it in the hierarchy than above it in the hierarchy. In order to achieve rapid reaction, not every input that comes into the lower levels can be passed along to the higher levels. Only the most important inputs are passed.

Goertzel illustrates this multilevel control structure by means of the three-level "pyramidal" vision processing parallel computer developed by Levitan and his colleagues at the University of Massachusetts. The bottom level deals with sensory data and with low-level processing such as segmentation into components. The intermediate level handles grouping, shape detection, and such; and the top level processes this information "symbolically," constructing an overall interpretation of the scene. This three-level perceptual hierarchy appears to be an extremely effective approach to computer vision.

> That orders are passed down the perceptual hierarchy was one of the biggest insights of the Gestalt psychologists. Their experiments (Kohler, 1975) showed that we look for certain configurations in our visual input. We look for those objects that we expect to see and we look for those shapes that we are used to seeing. If a level 5 process corresponds to an expected object, then it will tell its children [i.e., processes] to look for the parts corresponding to that object, and its children will tell their children to look for the complex geometrical forms making up the parts to which they refer, et cetera. (Goertzel 1994)

In his book *What Computers Can't Do*, Dreyfus points out that human intelligence is indivisible from the sense of presence in a body (see also Stuart 2003; Gärdenfors 2000, 2005). When we reason, we relate different ideas in a way that resembles the interrelations of parts of our body and the relation of our body with various external objects, which is in a complete agreement with the info-computational view, and the understanding of human cognition as a part of this overall picture.

## 3. Summary

In conclusion, let us sum up the proposed view of naturalized epistemology, based on the info-computationalist view of the universe.

> Within the info-computationalist framework, information is the stuff of the universe while computation is its dynamics. The universe is a network of computing processes and its phenomena are fundamentally info-computational in nature: as well continuous as discrete, analogue as digital computing are parts of the computing universe. On the level of quantum computing those aspects are inextricably intertwined. (Dodig-Crnkovic 2006)

Based on natural phenomena understood as info-computational, computing in general is conceived as an open interactive system (digital or analogue; discrete or continuous) in communication with the environment. The classical Turing machine is seen as a subset of a more general interactive/adaptive/self-organizing universal natural computer. A "living system" is defined as an "open, coherent, space-time structure maintained far from thermodynamic equilibrium by a flow of energy through it" (Chaisson 2002). On a computationalist view, organisms are constituted by computational processes, implementing computation in vivo. In the open system of living cells an info-computational process takes place using DNA, exchanging information, matter, and energy with the environment.

All cognizing beings are in constant interaction with their environment. The essential feature of cognizing living organisms is their ability to manage complexity and to handle complicated environmental conditions with a variety of responses that are results of adaptation, variation, selection, learning, and/or reasoning. As a consequence of evolution, increasingly complex living organisms arise. They are able to register inputs (data) from the environment, to structure those into information and, in more developed organisms, into knowledge. The evolutionary advantage of using structured, component-based approaches (data – information – knowledge) is improving response time and the computational efficiency of cognitive processes.

The main reason for choosing info-computationalist view for naturalizing epistemology is that it provides a unifying framework that makes it possible for different research fields such as philosophy, computer science, neuroscience, cognitive science, biology, and a number of others to communicate, exchange their results, and build a common knowledge.

It also provides the natural solution to the old problem of the role of representation in explaining and producing information, a discussion about two seemingly incompatible views: a symbolic, explicit, and static notion of representation versus an implicit and dynamic (interactive) one. Within the info-computational framework, those classical (Turing-machine type) and connectionist views are reconciled and used to describe different aspects of cognition.

The info-computationalist project of naturalizing epistemology by defining cognition as an information-processing phenomenon is based on the development of multilevel dynamical computational models and simulations of intelligent systems and has important consequences for the development of artificial intelligence and artificial life.

**Endnotes**

1. "Cellular automaton": "A regular spatial lattice of 'cells', each of which can have any one of a finite number of states. The states of all cells in the lattice are updated simultaneously and the state of the entire lattice advances in discrete time steps. The state of each cell in the lattice is updated according to a local rule that may depend on the state of the cell and its neighbors at the previous time step. Each cell in a cellular automaton could be considered to be a finite state machine that takes its neighbors' states as input and outputs its own state. The best known example is J.H. Conway's "Game of Life" (Free On Line Dictionary of Computing: http://foldoc.org/). For applications, see Wolfram 2002.
2. For an illustrative example, see http://dir.salon.com/story/tech/feature/2004/08/12/evolvable_hardware/index.html as quoted in Kurzweil 2005.
3. Normally this takes time, but there are obvious exceptions. Situations where the agent is in mortal danger are usually hard-coded and connected via a short-cut to activate an immediate, automatic, unconscious reaction. For a living organism, the efficiency of the computational process is presumably critical for its survival:

   "Over the billions of years of life on this planet, it has been evolutionarily advantageous for living organisms to be able to discern distinctions and patterns in their environment and then interact knowingly with that environment, based on the patterns perceived and formed. In the process of natural selection, those animals survive that are able to feed and reproduce successfully to the next generation. Being able to sense prey or predators and to develop strategies that protect one and promote the life success of one's offspring, these capabilities rest on a variety of forms of pattern detection, creation and storage. Consequently, organisms, particularly the higher animals, develop large brains and the skills to discern, cognitively process and operationally exploit information in the daily stream of matter and energy in which they find themselves. …In the broadest sense then, brains are buffers against environmental variability." (Bates 2005)
4. Here, a typical approach is connectionism, with the basic principle that mental phenomena are the emergent processes of interconnected networks of simple units. The most common forms of connectionism use neural network models. Learning is a basic feature of connectionist models. One of the dominant connectionist approaches today is Parallel Distributed Processing (PDP), which emphasizes the parallelism of neural processing and the distributed character of neural representations. It should be added that both connectionist and classical cognitive models are information processing and they both belong to the info-computationalist framework.
5. The name interactivism derives from the model for representation developed within this framework. Roughly, representation emerges in the presuppositions of anticipatory interactive processes in (natural or artificial) agents. The first dubbing of the model was by Rita Vuyk, who called it "Radical Interactivism" (Bickhard, "Interactivism: A Manifesto," at http://www.lehigh.edu/~mhb0/InteractivismManifesto.pdf).

**References**

Arnellos, A., T. Spyrou, and J. Darzentas. 2005. "The Emergence of Interactive Meaning Processes in Autonomous Systems." In *Proceedings of FIS 2005: Third International Conference on the Foundations of Information Science*. Paris, July 4-7, 2005.

Bates, M.J. "Information and Knowledge: An Evolutionary Framework for Information Science." *Information Research* 10 (2005). Accessible at http://InformationR.net/ir/10-4/paper239.html.

Berners-Lee, T., J. Hendler, and O. Lassila. "The Semantic Web." *Scientific American* 284 (2001): 34-43. Accessible at http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&ref=sciam.

Bickhard, M.H. 2004. "The Dynamic Emergence of Representation." In *Representation in Mind: New Approaches to Mental Representation*, edited by H. Clapin, P. Staines, P. Slezak. 71-90. Oxford: Elsevier Ltd.

Burgin, M. 2005. *Super-Recursive Algorithms (Monographs in Computer Science)*. New York: Springer.

Campbell, D.T. and B.T. Paller. 1989. "Extending Evolutionary Epistemology to "Justifying Scientific Beliefs (A sociological rapprochement with a fallibilist perceptual foundationalism?).” In *Issues in Evolutionary Epistemology*, edited by K. Hahlweg and C.A. Hooker. 231-57. Albany: State University of New York Press.

Chaisson, E.J. 2001. *Cosmic Evolution. The Rise of Complexity in Nature*. 16-78. Cambridge, MA: Harvard University Press.

Chaitin, G. 2006. “Epistemology as Information Theory.” *COLLAPSE* Vol. I. 27-51. Alan Turing Lecture given at E-CAP 2005. Accessible at http://www.cs.auckland.ac.nz/CDMTCS/chaitin/ecap.html.

Coffman, A.J. “Developmental Ascendency: From Bottom-up to Top-down Control.” *Biological Theory Spring* 1 (2006): 165-78.

Dawson, M.R.W. 2006. “Understanding Cognitive Science.” Oxford: Blackwell Publishers.

Dodig-Crnkovic, G. 2006. “Investigations into Information Semantics and Ethics of Computing.” Sweden: Mälardalen University Press. Accessible at http://www.diva-portal.org/mdh/abstract.xsql?dbid=153.

Dreyfus, H.L. 1972. *What Computers Can’t Do: A Critique of Artificial Reason*. New York: Harper & Row.

Floridi, L. “What is the Philosophy of Information?” *Metaphilosophy* 33 (2002): 123-45.

Flake, G. W. 1998. *The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation*. Cambridge, MA: MIT Press.

Gell-Mann, M. 1995. *The Quark and the Jaguar: Adventures in the Simple and the Complex*. New York: Owl Books.

Goertzel, B. 1993. “The Evolving Mind.” Gordon and Breach.

———. 1994. *Chaotic Logic*. Plenum Press.

Gärdenfors, P. 2003. *How Homo Became Sapiens: On the Evolution of Thinking*. Oxford: Oxford University Press.

———. 2000. *Conceptual Spaces*. Cambridge, MA: Bradford/MIT Press.

———, J. Zlatev, and T. Persson. 2005. “Bodily mimesis as ‘the missing link’ in human cognitive evolution.” Lund University Cognitive Studies 121, Lund.

Harms, W.F. 2004. *Information and Meaning in Evolutionary Processes*. Cambridge: Cambridge University Press.

Harms, W.F. “Naturalizing Epistemology: Prospectus 2006,” *Biological Theory* 1 (2006): 23-24.

Kornblith, H. ed. 1994. *Naturalizing Epistemology*, 2nd ed. Cambridge: MIT Press.

———. “Knowledge in Humans and Other Animals.” *Noûs* 33 (1999): 327.

Kulakov, A. and G. Stojanov. 2002. “Structures, Inner Values, Hierarchies and Stages: Essentials for Developmental Robot Architecture.” 2nd International Workshop on Epigenetic Robotics. Edinburgh.

Kurzweil, R. 2005. *The Singularity is Near*. New York: Viking.

Lloyd, S. 2006. *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*. New York: Alfred A. Knopf.

Lorenz, K. 1977. *Behind the Mirror*. London: Methuen.

MacLennan, B. “Natural Computation and Non-Turing Models of Computation,” *Theoretical Computer Science* 317 (2004): 115-45.

Maturana, H. and F. Varela. 1992. *The Tree of Knowledge*. Shambala.

———. 1980. *Autopoiesis and Cognition: The Realization of the Living*. Boston:D. Reidel.

Piaget, J. 1987. *Possibility and Necessity*, Vols. 1 and 2. Minneapolis: University of Minnesota Press.

Popper, K.R. 1972. *Objective Knowledge: An Evolutionary Approach*. Oxford: Clarendon/Oxford University Press.

Stuart, S. “The Self as an Embedded Agent,” *Minds and Machines* 13 (2003): 187.

Toulmin, S. 1972. *Human Understanding: The Collective Use and Evolution of Concepts*. Princeton, NJ: Princeton University Press.

Wegner, P. “Interactive Foundations of Computing,” *Theoretical Computer Science* 192 (1998): 315-51.

Whitehead, A.N. 1978. *Process and Reality: An Essay in Cosmology*. New York: The Free Press.

Wolfram, S. 2002. “A New Kind of Science.” Wolfram Media. Accessible at http://www.wolframscience.com.

---

# *Taking the Intentional Stance Toward Robot Ethics*

**James H. Moor**
*Dartmouth College*

I wish to defend the thesis that robot ethics is a legitimate, interesting, and important field of philosophical and scientific research. I believe it is a coherent possibility that one day robots will be good ethical decision-makers at least in limited situations and act ethically on the basis of their ethical understanding. Put another way, such envisioned future robots will not only act *according to* ethical principles but act *from* them.

This subject goes by various names such as “robot ethics,” “machine ethics,” or “computational ethics.” I am not committed to any particular term, but I will here use “robot ethics” as it suggests artificial agency. I do not exclude the possibility of a computer serving as an ethical advisor as part of robot ethics, and I include both software and hardware agents as candidates for robots.

## Kinds of Ethical Robots

Agents, including artificial agents, can be understood as ethical in several ways. I distinguish among at least four kinds of ethical agents (Moor 2006). In the weakest sense *ethical impact agents* are simply agents whose actions have ethical consequences whether intended or not. Potentially any robot could be an ethical impact agent to the extent that its actions cause harms or benefits to humans. A computerized watch can be considered an ethical impact agent if it has the consequence of encouraging its owner to be on time for appointments. The use of robotic camel jockeys in Qatar has the effect of reducing the need for slave boys to ride the camels.

*Implicit ethical agents* are agents that have ethical considerations built into their design. Typically, these are safety or security considerations. Planes are constructed with warning devices to alert pilots when they are near the ground or when another plane is approaching on a collision path. Automatic teller machines must give out the right amount of money. Such machines check the availability of funds and often limit the amount that can be withdrawn on a daily basis. These agents have designed reflexes for situations requiring monitoring to ensure safety and security. Implicit ethical agents have a kind of built in virtue—not built from habit but from specific implementations in programming and hardware.

Unethical agents exist as well. Moreover, some agents can be ethical sometimes and unethical at others. One example of such a mixed agent I will call “the Goodman agent.” The Goodman agent is an agent that contains the millennium bug. This bug was generated by programming yearly dates using only the last two digits of the number of the year resulting in dates beyond 2000 being regarded as existing earlier than those in the late 1900s. Such an agent was an ethical impact agent before 2000 and an unethical impact agent thereafter. *Implicit unethical agents* exist as well. They have built in vice. For instance, a spam zombie is an implicit unethical agent. A personal computer can be transformed into a spam zombie if it is infected by a virus that configures the computer to send spam e-mail to a large number of victims.

Ethical impact agents and implicit ethical agents are ethically important. They are familiar in our daily lives, but there is another kind of agent that I consider more central to robot ethics. *Explicit ethical agents* are agents that can identify and process ethical information about a variety of situations and make sensitive determinations about what should be done in those situations. When principles conflict, they can work out resolutions that fit the facts. These are the kind of agents that can be thought of as acting from ethics, not merely according to ethics. Whether robot agents can acquire knowledge of ethics is an open empirical question. On one approach ethical knowledge might be generated through good old-fashioned AI in which the computer is programmed with a large script that selects the kinds of information relevant to making ethical decisions and then processes the information appropriately to produce defensible ethical judgments. Or the ethical insights might be acquired through training by a neural net or evolution by a genetic algorithm. Ethical knowledge is not ineffable and that leaves us with the intriguing possibility that one day ethics could be understood and processed by a machine.

In summary, an ethical impact agent will have ethical consequences to its actions. An implicit ethical agent will employ some automatic ethical actions for fixed situations. An explicit ethical agent will have, or at least act as if it had, more general principles or rules of ethical conduct that are adjusted or interpreted to fit various kinds of situations. A single agent could be more than one type of ethical agent according to this schema. And the difference between an implicit and explicit ethical agent may in some cases be only a matter of degree.

I distinguish explicit ethical agents from full ethical agents. *Full ethical agents* can make ethical judgments about a wide variety of situations and in many cases can provide some justification for them. Full ethical agents have those metaphysical features that we usually attribute to ethical agents like us, features such as intentionality, consciousness, and free will. Normal adult humans are our prime examples of full ethical agents. Whether robots can become full ethical agents is a wonderfully speculative topic but not one we must settle to advance robot ethics. My recommendation is to treat explicit ethical agents as the paradigm example of robot ethics. These potential robots are sophisticated enough to make them interesting philosophically and important practically. But not so sophisticated that they might never exist.

An explicit ethical robot is futuristic at the moment. Such activity is portrayed in science fiction movies and literature. In 1956, the same year of the Summer Project at Dartmouth that launched artificial intelligence as a research discipline, the movie "Forbidden Planet" was released. A very important character in that movie is Robby, a robot that is powerful and clever. But Robby is merely a robot under the orders of human masters. Humans give commands and he obeys. In the movie we are shown that his actions are performed in light of three ethical laws of robotics. Robby cannot kill a human even if ordered to do so.

Isaac Asimov had introduced these famous three laws of robotics in his own short stories. Asimov's robots are ethical robots, the kind I would characterize as explicit ethical agents. They come with positronic brains that are imbued with the laws of robotics. Those who are familiar with Asimov's stories will recall that the three laws of robotics appear in the *Handbook of Robotics*, 56th Edition, 2058 A.D. (Asimov 1991):

1. A robot may not injure a human being, or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

Asimov's robots are designed to consult ethical guidelines before acting. They are kind and gentle robots compared to the terrifying sort that often appears in books and movies. Asimov's ethical laws of robotics seem reasonable at least initially, but, if pursued literally, they are likely to produce unexpected results. For example, a robot, which we want to serve us, might be obligated by the first law to travel into the world at large to prevent harm from befalling other human beings. Or our robot might interfere with many of our own plans because our plans for acting are likely to contain elements of risk of harm that needs to be prevented on the basis of the first law.

Although Asimov's three laws are not adequate as a system of ethics for robots, the conception that Asimov was advancing seems to be that of a robot as an explicit ethical agent. His robots could reason from ethical principles about what to do and what not to do. His robots are fiction but they provide a glimpse of what it would be like for robotic ethics to succeed.

## Evaluating Explicit Ethical Robots

I advocate that we adopt an empirical approach to evaluating ethical decision making by robots (Moor 1979). It is not an all or nothing matter. Robots might do well in making some ethical decisions in some situations and not do very well in others. We could gather evidence about how well they did by comparing their decisions with human judgments about what a robot should do in given situations or by asking the robots to provide justifications for their decisions, justifications that we could assess. Because ethical decision making is judged by somewhat fuzzy standards that allow for disagreements, the assessment of the justification offered by a robot for its decision would likely be the best and most convincing way of analyzing a robot's ethical decisions competence. If a robot could give persuasive justifications for ethical decisions that were comparable to or better than that of good human ethical decision makers, then the robot's competence could be inductively established for a given area of ethical decision making. The likelihood of having robots in the near future that are competent ethical decision makers over a wide range of situations is undoubtedly small. But my aim here is to argue that it is a coherent and defensible project to pursue robot ethics. In principle we could gather evidence about their ethical competence.

Judging the competence of a decision maker is only part of the overall assessment. We need also to determine whether it is appropriate to use the decision maker in a given situation. A robot may be competent to make a decision about what some human should have for her next meal. Nevertheless, she would probably justifiably wish to decide for herself. Therefore, a robot could be ethically competent in some situations in which we would not allow the robot to make such decisions because of our own values. With good reason we usually do not allow other adults to make ethical decisions for us, let alone allow robots to do it. However, it seems possible there could be specific situations in which humans were too biased or incompetent to be fair and efficient. Hence, there might be a good *ethical* argument for using a robotic ethical decision maker in their place. For instance, a robotic decision maker might be more competent and less biased in distributing assistance after a national disaster like the hurricane Katrina that destroyed much of New Orleans. In the Katrina case the human relief effort was incompetent. The coordination of information and distribution of goods was not handled well. In the future ethical robots might do a better job in such a situation. Robots are spectacular at tracking large amounts of information and could communicate with outlets to send assistance to those who need it immediately. These robots might at some point have to make triage decisions about whom to help first, and they might do this more competently and fairly than humans. Thus, it is conceivable there could be persuasive *ethical* arguments to employ robot ethical decision makers in place of human ones in selected situations.

### The Intentional Stance

I have selected robots that are explicit ethical agents as the interesting class of robots for consideration in robot ethics. Of course, if robots one day become persons and thereby full ethical agents, that would be even more interesting. But that day is not likely to come in the foreseeable future, if at all. Nonetheless, explicit ethical agents, though not full ethical agents, could be quite sophisticated in their operations. We might understand them by regarding them in terms of what Daniel Dennett calls "the intentional stance" (Dennett 1971). In order to predict and explain the behavior of complex computing systems, it is often useful to treat them as intentional systems. To treat them as if they were rational creatures with beliefs and desires pursuing goals. As Dennett suggests, predicting and explaining computer behavior on the basis of the *physical stance* using the computer's physical makeup and the laws of nature or on the basis of the *design stance* using the functional specifications of the computer's hardware and programming is useful for some purposes such as repairing defects. But predicting and explaining the overall behavior of computer systems in terms of the physical and the design stances is too complex and cumbersome for many practical purposes. The right level of analysis is in terms of the intentional stance.

Indeed, I believe most computer users often take the intentional stance about a computer's operations. We predict and explain its actions using the vocabulary of beliefs, desires, and goals. A word processing program corrects our misspellings because it *believes* we should use different spellings and its *goal* is to correct our spelling errors. Of course, we need not think the computer believes or desires in the way we do. The intentional stance can be taken completely instrumentally. Nevertheless, the intentional stance is useful and often an accurate method of prediction and explanation. That is because it captures in a rough and ready way the flow of the information in the computer. Obviously, there is a more detailed account of what the word processing program is doing in terms of the design stance and then at a lower level in terms of the physical stance. But most of us do not know the details nor do we need to know them in order to reliably predict and explain the word processing program's behavior. The three stances (intentional, design, and physical) are consistent. They differ in level of abstraction.

We can understand robots that are explicit ethical agents in the same way. Given their beliefs in certain ethical principles, their understanding of the facts of certain situations, and their desire to perform the right action, they will act in such and such ethical manner. We can gather evidence about their competence or lack of it by treating them as intentional systems. Are they making appropriate ethical decisions and offering good justifications for them? This is not to deny that important evidence about competence can be gathered at the design level and the physical level. But an overall examination and appreciation of a robot's competence is best done at a more global level of understanding.

### Why Not Ethical Robots Now?

What prevents us from developing ethical robots? Philosophically and scientifically is the biggest stumbling block metaphysical, ethical, or epistemological?

Metaphysically, the lack of consciousness in robots seems like a major hurdle. How could explicit ethical agents really do ethics without consciousness? But why is consciousness necessary for doing ethics? What is crucial is that the robot receives all of the necessary information and processes it in an acceptable manner. A chess playing computer lacks consciousness but plays chess. What matters is that the chess program receives adequate information about the chess game and processes the information well so that by and large it makes reasonable moves.

Metaphysically, the lack of free will would also seem to be a barrier. Don't all moral agents have free will? For sake of argument let's assume that full ethical agents have free will and robots do not. Why is free will necessary for acting ethically? The concern about free will is often expressed in terms of a concern about human nature. A common view is that humans have a weak or base nature that must be overcome to allow them to act ethically. Humans need to resist temptations and self-interest at times. But why do robots have to have a weak or base nature? Why can't robots be built to resist temptations and self-interests when it is inappropriate? Why can't ethical robots be more like angels than us? We would not claim a chess program could not play championship chess because it lacks free will. What is important is that the computer chess player can make the moves it needs to make in the appropriate situations as causally determined as those moves may be.

Ethically, the absence of an algorithm for making ethical decisions seems like a barrier to ethical robots. Wouldn't a computer need an algorithm to do ethics (Moor 1995)? Let us assume there is no algorithm for doing ethics, at least no algorithm that can tell us in every situation exactly what we should do. But, if we act ethically and don't need an algorithm to do it, we do it in some way without an algorithm. Whatever our procedure is to generate a good ethical decision, why couldn't a robot have a similar procedure? Robots don't have to be perfect to be competent any more than we do. Computers often have procedures for generating acceptable responses even when there is no algorithm to generate the best possible response.

Ethically, the inability to hold the robot ethically responsible seems like a major difficulty in pursuing robot ethics. How would we praise or punish a robot? One possibility is that robots might learn like us through some praise or punishment techniques. But a more direct response is that ethical robots that are not full ethical agents would not have rights, and could be repaired. We could hold them causally responsible for their actions and then fix them if they were malfunctioning so they act better in the future.

Epistemologically, the lack of ability of robots to have empathy for humans would lead them to overlook or not appreciate human needs. This is an important insight as much of our understanding of other humans depends on our own emotional states. Of course, we might be able to give robots emotions, but short of that we might be able to compensate for their lack of emotions by giving them a theory about human needs including behavioral indicators for which to watch. Robots might come to know about emotions by other means than feeling the emotions. A robot's understanding of humans might be possible through inference if not directly through emotional experience.

Epistemologically, computers today lack much common sense knowledge. Hence, robots could not do ethics, which so often depends upon common sense knowledge. This is probably the most serious objection to robot ethics. Computers work best in well-defined domains and not very well in open environments. But robots are getting better. Autonomous robotic cars are adaptable and can travel on most roads and even across open deserts and through mountain tunnels when given the proper navigational equipment. Robots that are explicit ethical agents lacking common sense knowledge would not do as well as humans in many settings but might do well enough in a limited set of situations. In some cases, such as the example of the disaster relief robot, that may be all that is needed.

## Conclusion

We are some distance from creating robots that are explicit ethical agents. But this is a good area to investigate scientifically and philosophically. Aiming for robots that are full ethical agents is to aim too high at least for now, and to aim for robots that are implicit ethical agents is to be content with too little. As robots become increasingly autonomous, we will need to build more and more ethical considerations into them. Robot ethics has the potential for a large practical impact. In addition, to consider how to construct an explicit ethical robot is an exercise worth doing for it forces us to become clearer about what ethical theories are best and most useful. The process of programming abstract ideas can do much to refine them.

### References

Asimov, Isaac. 1991. *Robot Visions*. New York: Penguin Books.

Dennett, Daniel. "Intentional Systems," *Journal of Philosophy* LXVIII (1971): 87-106.

Moor, James H. "Are There Decisions Computers Should Never Make?" *Nature and System* 1 (1979): 217-29.

Moor, James H. "Is Ethics Computable?" *Metaphilosophy* 26 (January/April 1995): 1-21.

Moor, James H. "The Nature, Importance, and Difficulty of Machine Ethics," *IEEE Intelligent Systems* 21 (July/August 2006): 18-21.

---

# *Open Source Software and Consequential Responsibility: GPU, GPL, and the No Military Use Clause*

**Keith W. Miller**
*University of Illinois at Springfield*

## Introduction

Much has been written about open source software ("OSS") and its ethical implications, and this is still an active area. For example, there was a recent call for papers for a special journal issue on this topic (*Journal of the European Ethics Network*). This paper focuses on one narrow issue that has arisen with respect to OSS: whether or not OSS developers should control or have responsibility for how their software is used by others after its release.

This paper is an examination of this issue through the lens of an incident regarding an OSS project called "GPU." At one point in GPU's development, its developers attempted to add a clause to their license based on Asimov's First Law of Robotics (Asimov 1950). The GPU website characterized this restriction as a "no military use" clause. Eventually the GPU developers rescinded this license restriction under pressure because the restriction violated the requirements of two important OSS organizations.

This paper begins with a brief history of OSS and two organizations that help define and promote OSS. The remainder of the paper describes the GPU case in some detail and presents several arguments about its ethical ramifications.

## A Brief History of Open Source Software

Many of the events discussed here are described in Wikipedia (Open-source). In 1955, shortly after the commercial release of IBM's first computer, the organization SHARE was formed by customers interested in the source code of the IBM operating system. IBM made the source code of operating systems available in the SHARE library, as well as modifications made by users. The president of SHARE stated, "SHARE and its SHARE library invented the open source concept" (SHARE).

SHARE may have been the first organization to formalize source code sharing, but it didn't formalize the term "open source." That term gained popularity after a "strategy session" in 1998 that was called in anticipation of the public distribution of the source code for the Netscape browser (OSI, History). One of the reasons this group adopted the term "open source" was to distinguish the concept from "free software" as defined by Richard Stallman and the Free Software Foundation ("FSF") founded in 1985 (FSF, GNU). The differences between open source and the Open Source Institute ("OSI") on the one hand, and free software and the FSF on the other hand, have been a major source of contention and public discussion. Some people now use the inclusive terms FOSS ("Free and Open Source Software") and FLOSS ("Free/Libre/Open Source Software") to identify both camps in the debate. This paper will use FLOSS as the collective term.

FSF and OSI are important to the GPU case. These two organizations do not embody all possible or existing ideas about what OSS is or should be. However, both have issued influential, public descriptions of their open source philosophies, and their pronouncements about what does and does not qualify as OSS have discernible effects in this case.

## How FSF and OSI Describe FLOSS

The web sites for FSF and OSI include explicit statements about the theory behind their practice of FLOSS. This section gives a sampling of the language each web site uses that is relevant to the ethical analyses of this paper. In addition to materials of the three organizations, this section also relies on a survey found in Grodzinsky et al. (2003).

Both FSF and OSI are in some sense a counterpoint to software developers who release only object code instead of source code, and who use licensing, copyright, and patent laws to restrict access to their source code and, in some cases, the object code. Following common definitions, we will call this non-FLOSS software "proprietary."

## FSF

Richard Stallman was part of a programming elite at MIT in the early 1970s. Stallman participated in a culture that freely exchanged, modified, and reused source code. In 1984, Stallman began the GNU project, hoping to recreate a culture that emphasized sharing software.

FSF grew out of the GNU project. FSF advocates "free software." Stallman is careful to define what "free" means in this context, famously making this comment: "Free software is a matter of liberty, not price. To understand the concept, you should think of free as in free speech, not as in free beer" (FSF, Free software). Free software is further defined with the "four freedoms," again from the FSF website:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

In order to perpetuate the four freedoms, FSF uses and advocates "copyleft." FSF (Copyleft) states:

> To copyleft a program, we first state that it is copyrighted; then we add distribution terms, which

> are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code *or any program derived from it* but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable. Proprietary software developers use copyright to take away the users' freedom; we use copyright to guarantee their freedom. That's why we reverse the name, changing "copyright" into "copyleft."

The copyleft idea was formalized into a General Public Licence, or GPL. FSF requires that anyone using GNU software accept the GPL, and attach the GPL to any software derived from GNU software.

The GPL, some of the FSF documents, and Stallman's sometimes dramatic rhetoric in talks and interviews have not been accepted universally, even by programmers enthusiastic about OSS. Some of these programmers have become involved in OSI, an organization separate from FSF, but also dedicated to promoting OSS.

### OSI

OSI traces its history to 1998, when Netscape announced that it was releasing the source code of its browser. Soon after that announcement, a "brain storming session" in Palo Alto started using the term "open software," which became a competitor to Stallman's vision of "free software." Users and developers of Linux and Netscape began referring to their source code as "open," and OSI was formally begun in 1998 (OSI, History).

The OSI site includes links to articles that discuss its philosophical foundations. One such link leads to an article by Prasad (2001) that includes the following: "Open Source is doing what god, government and market have failed to do. It is putting powerful technology within the reach of cash-poor but idea-rich people."

Although these value-laden discussions do occur in relation to OSI and its website, most of the discussion by OSI is more pragmatic and oriented towards arguments about why FLOSS is more stable, less brittle, and economically advantageous as compared to proprietary software. The major thrust is reflected in this excerpt from the OSI homepage (OSI Welcome):

> The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, and people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.
>
> We in the open source community have learned that this rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits.
>
> Open Source Initiative exists to make this case to the commercial world.

### FSF Advocates GPL and OSI Accepts GPL

FSF advocates a single license, the GPL, and allows two: GPL and LGPL (the "lesser" GPL, not described in this paper). OSI has a more inclusive strategy. OSI (Definition 2006) publishes a definition of what an open source license should allow and not allow and maintains a list of OSI-approved licenses that fit its definition (OSI Licensing 2006). At this writing there are over fifty OSI-approved licenses, including GPL and LGPL. This means that OSI approves of GPL and LGPL licensed software as officially "open source software."

FSF does not include OSI in its list of free software organizations because the OSI definition includes licenses that do not fit the FSF definition of "free software." FSF (Links) does, however, list OSI in a list of "organizations related to free software."

One reason that this background is useful in understanding the GPU case is that despite their many differences, FSF and OSI agreed on the proper response to the GPU attempt at restricting its license. Both FSF and OSI opposed the GPU attempted license restriction, and both oppose any such restrictions.

### GPU and the No Military Use License Patch

GPU (not to be confused with FSF's GNU project) is software designed to link PCs together in order to share CPU cycles (GPU). The authors of the software originally released GPU with the GPL license, and the project was "hosted" by SourceForge.net, a website for FLOSS. The GPU website's news archive has announcements that stretch back to 2002.

A GPU announcement on August 31, 2005, a "status report," included paragraphs about "Look & Feel" and "Improvements to whiteboard." It also included this:

> GPL for no military use
>
> Following inquires of Nick Johnson (npj), we decided to create our own version of the GPL. The text can be read here http://gpu.sourceforge.net/GPL_license_modified.txt. Goal of the modification is to prohibit military use, without giving away the advantages provided by GPL

Almost a year later, on August 14, 2006, NewsForge, "an online newspaper for Linux and Open Source," published an article about GPU's no military use modification of the GPL (Gasperson 2006). That article attracted a great deal of attention to GPU and its altered license. On the same day, the following announcement was on the GPU website. (Grammar and spelling mistakes are retained from the original announcements below.)

> Discussion about modified GPL
>
> What started like a little taunt suddenly got another dimension. The GPU project has modified the GPL license a little by adding Asimov's first law of robotics.
>
> Meanwhile, we have been written be members of the Free Software Foundation, asking us to reconsider the change or at least not violate their copyright by removing the preamble and altering the name. We are aware modifying the GPL is not allowed by the GPL license itself, but did it without bad intentions. We go consider what is appropriate. After all, we're not after a legal conflict with the FSF. Give us some time for internal debate, we'll keep you informed.

Five days later, the GPU website had this announcement:

> 0.935 Project reverts to plain GPL
>
> After an internal discussion between team members, we decided to release 0.935 with the unmodified (GPL version 2), and to remove the public released versions beginning from 0.910 up to 0.934.
>
> This for two main reasons: one is that Sourceforge.net hosts only projects that are licensed under the Open

> Source Definition. The project is not big enough to provide its own CVS and web space.
>
> The second one is that GPL cannot be modified without changing its name. So we should have chosen a name like "No military Use Public License."
>
> There was discussion going on for the GPL version 3 that regards a restriction for military use. Read for example David Turner's blog: http://www.fsf.org/blogs/licensing/20050211.html.
>
> Release 0.935 includes a new search plugin and frontend by nanobit, an updated DelphiPackagerTool by DelphiFreak and an attempt to include JPL planets inside Orsa.

The GPU no military use clause has been discussed in hundreds of websites, but few of these sites include the information that the clause has been retracted.

The rest of this paper will focus on two issues dramatized by the GPU case. First, we will examine the reasons that both FSF and OSI oppose the kind of restriction that GPU was attempting. Second, we will explore a broader, related issue, a challenge to FSF and OSI based on their refusal to allow such restrictions.

### FSF and OSI Oppose Use Clauses

The defining documents of FSF and OSI prohibit license restrictions on the use of OSS. OSI (Definition) includes the following two clauses in its definition of OSS:

> 5. No Discrimination Against Persons or Groups
>
> The license must not discriminate against any person or group of persons.
>
> Rationale: In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore, we forbid any open-source license from locking anybody out of the process.
>
> Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.
>
> 6. No Discrimination Against Fields of Endeavor
>
> The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.
>
> Rationale: The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.

The FSF GPL includes language with a similar effect: "You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License." Stallman's first freedom (freedom 0, listed above) speaks directly to this issue: "The freedom to run the program, for any purpose."

The FSF website (Turner 2005) includes the following under the title "Censorship Envy and Licensing":

> So, we reject restrictions on who can use free software, or what it can be used for. By keeping everyone on a level playing field, we get the widest possible participation in the free software movement. And the anti-nuclear activists are still free to use free software to organize and spread their ideas.

The OSS movement, as represented by FSF and OSI, prohibits software developers from trying to restrict via licensing how people use FLOSS they develop. Is this prohibition ethical? At least some suggest that the prohibition is not ethical and that FLOSS developers have an obligation with respect to how their software is used by others.

### A Computer Ethics Argument about License Restrictions

When the GPU license restriction became known through the NewsForge article, many Web comments were posted that supported the idea of restricting open source software from military use; others supported the open nature of FLOSS that prohibits such restrictions. For examples of these arguments, see Yesh (2006) and Klepas (2006).

Interestingly, the controversy over the GPU military use clause was foreshadowed in the computer ethics literature. For a 2002 panel titled "Open source software: intellectual challenges to the status quo" (Wolf et al. 2002), Don Gotterbarn wrote the following.

> …the OSS standard says "The license must not restrict anyone from making use of the program in a specific field of endeavor." This means that when I make a piece of software Open Source, I lose the right to control the ethical and moral impact of what I have created. Being forced to abrogate this right is not acceptable…

The phrase "forced to abrogate" is apt in the case of the GPU case. If the GPU project had insisted on retaining the military use clause, GPU would have had to abandon its claim to a GPL and its claim to be compliant with OSI. GPU could not have remained an official Open Source project and would have lost its Web home at SourceForge.net. Judging by the timing and the wording of its retraction of the military use clause, these considerations were pivotal.

### Responding to Gotterbarn's "Right to Control the Ethical and Moral Impact"

In this section we examine Gotterbarn's claim that making software available with an open source license requires that the developer give up the right to control the ethical and moral impact of what was developed. First, the idea that a developer is "losing" or "abrogating" this right presupposes that the developer had the right before releasing the software. Reasoning first by analogy, it is not clear that anyone writing software has such a right. A person who produces something for public consumption is rarely assumed to have such a far-reaching right, nor are they likely to claim it. Someone who produces a hammer or a mousetrap and offers it for sale does not expect to exercise control over who might buy the item or what they might do with it after it is bought. By this reasoning, if the GPU developers have such a right, it must be because of some special nature of their software.

There are exceptions to this rule about controlling who can use something you produce. Someone who produces weapons or potentially dangerous biological agents likely expects that some government agency will restrict who might receive these products. Producers of cigarettes, alcohol, and some forms of entertainment also expect that minors will be prohibited from buying their products. Restrictions have been attempted regarding what countries can obtain encryption software (Zimmerman). In all these cases, the producer is restricted from selling the products to certain potential markets. The government is adding an obligation to the seller so that the seller must restrict the set of buyers. These legal provisions suggest that this kind of restriction is an obligation rather than the enforcement of a seller's right. Indeed, in many transactions, there are laws prohibiting sellers from discriminating against buyers on the basis of race, gender, or creed. (Think of, for example, prohibitions against discriminatory practices in selling real estate [US Code 2006].)

If there is not an inherent right to control what users do with something you produce, there might still be an ethical obligation to attempt such control. Gotterbarn, for example, favors licensing that adds "some control over use" (Wolf et al. 2002b). The idea that ethical responsibility for software reaches beyond release of the software has intuitive appeal, especially if the authors can clearly foresee potentially dangerous or injurious uses of the software they release. For example, if a developer produced software that quickly broke an existing encryption algorithm that was being widely used for secure communications, the consequences of a sudden, public release of that software without notification to the public that such a release was forthcoming would have foreseeable, significant consequences. A plausible consequential argument can be constructed that a programmer would have ethical, if not legal, responsibilities for releasing that software carefully if at all.

Some could argue that the release would be justified because of the negative consequences of secrecy, or because dependence on an encryption algorithm that had been proven vulnerable would also be dangerous. That is an interesting debate, but it will not be pursued further in this paper. Instead, we will state that in any case when significant, direct consequences of software are reasonably foreseeable before the release of software, a programmer has an ethical responsibility to consider those consequences before releasing the software.

There are problems, however, with using the existence of this consequential argument to conclude that the OSI clause six and FSF's freedom 0 are, therefore, ethically invalid. (The remainder of this paper will use "clause six" to refer to both the OSI language and the FSF language.) Clause six does not prohibit a programmer from considering eventual uses before releasing software; clause six merely prohibits writing a FLOSS license in such a way that it restricts subsequent uses. The programmer can certainly decide not to release the software at all, in which case clause six is irrelevant. Or the developers could decide to release the code with a modified GPL license that included restrictive clauses. Although there is some question as to whether or not the restrictive clause would be effective (more on this later), the GPU developers are certainly free to include such clauses in their license as long *as they do not then claim that their software is OSI or GPL compliant*. If GPU wants to be FLOSS, then they must submit to the FLOSS rules. An obligation for a software developer to use restrictions in order to attempt to affect consequences does not automatically transfer an obligation on OSI or FSF to facilitate those restrictions.

Even though a developer's obligations do not automatically transfer to OSI and FSF, perhaps the existence of some cases in which license restrictions are ethically necessary could be part of an argument that ultimately places an obligation on OSI and FSF to permit such restrictions. One could argue that clause six discourages proactive, ethical actions of open source developers since an open source developer is blocked from licensing software for "good uses" while prohibiting "bad uses."

There are both practical and theoretical objections to claiming that clause six is, therefore, unethical. First, even if an open source release could include a license prohibition like the military use ban, there is no reasonably effective method of enforcing the ban against the prohibited use. Once source code is released openly, the proverbial cat is out of the bag. If the use occurs there may be legal ramifications after the fact, but it is not at all clear that the case will be successful. For one thing, the methods of obfuscating borrowed code are numerous. Also, the legal decision of whether or not a particular reuse of the software falls into the description of "good" and "bad" uses will certainly be difficult to assess in many situations. And any such legal wrangling will have to happen after the "bad use" has already occurred, suggesting that the ethical responsibility will not have been fulfilled despite the language in the license.

There may be a symbolic and useful effect from attempting restrictions (more on this later), but it is difficult to ensure that these kinds of actions will be otherwise effective. The GPU developers' announcement that refers to their military use restriction as "a little taunt" is consistent with our contention that the restriction was unlikely to be effective in more than a symbolic way.

A second objection against changing clause six to allow specific reuse prohibitions is that allowing such prohibitions opens a Pandora's box that would seriously erode the usefulness of open source software. Turner (2001) makes this same argument. If developers routinely add such stipulations, then they will either be ignored (rendering them useless), or else they will be observed, leading to complications and inefficiencies similar to the problems associated with software patents (League for Programming Freedom). Neither of these outcomes is likely to significantly reduce "bad uses" of the software, but they are likely to impede the good consequences of open software. The advantages of FLOSS, argued by FSF, OSI, and others (including other panelists in the session where Gotterbarn made his objections) would, therefore, be at risk if licensing restrictions like the military use clause were allowed.

Another objection against a general obligation for software developers to include reuse prohibitions in open software licenses (or proprietary licenses, for that matter) is the difficulty of anticipating how a particular piece of software will be used after its release. While the decryption algorithm above would have obvious uses that might cause concern, other pieces of software are far more general in nature. For example, if software renders pictures or sorts lists, the possible uses are endless. Requiring an open source developer to anticipate, describe, and prohibit any future use deemed to be an ethical wrong seems both futile and unreasonable. Many high visibility, widely distributed FLOSS projects focus on utility applications such as Web servers, operating systems, and programming language implementations. In all three of these cases, the eventual uses are so numerous and diverse that trying to anticipate future uses with any accuracy is futile. Furthermore, if restrictions against the use of these utilities were effective (again, we have argued that this is unlikely), users have many non-FLOSS options available, so that restricting the FLOSS utility would not be likely to halt the anticipated bad use unless no alternatives existed.

If specific uses related to the nature of the software cannot be accurately predicted, this leads us to conclude that except in extraordinary cases, license restrictions will not target uses of the software as much as it will target specific users or classes of users. When restrictions target users instead of uses, developers are adopting the role of judge and jury on classes of potential users. This seems a heavy ethical burden indeed on software developers. If developers want to take on this burden, they can do so by stepping outside the FLOSS movement. However, it does not seem appropriate to do this kind of judging under the banner of "open software." When the restrictions are based on a judgment of potential users, the explicit goal of the restrictions is closing the software to certain people, not opening up the community of users and developers.

Finally, it seems arbitrary to assign an ethical obligation on open source developers to anticipate and block unethical uses of FLOSS when no such ethical obligation has been required of non-FLOSS developers. It is easier to critique the licensing requirements of open source FLOSS because those requirements are clearly and publicly stated, at least with respect to FSF and OSI. The licensing requirements of proprietary software are far more numerous and often less accessible to the public (especially for private contract software). However, neither the open nature of open source licenses nor the more private nature of proprietary software licenses should affect the vulnerability of either type of software to accountability arguments. If open source software developers are accountable for subsequent uses, then so are proprietary software developers. Singling out FLOSS developers alone for this obligation is unfair.

## The Power of GPU's Symbolic Act

We have argued above that it is unlikely that restrictions such as GPU's no military use clause would have more than a symbolic effect. However, symbolic acts can have important consequences. Certainly GPU's attempted restriction stirred up active discussions, at least among FLOSS developers and users, about military uses of software, about ethical responsibilities of software developers for the consequences of their work, and about possible ethical obligations of FSF and OSI. This consequence can be seen as positive and, therefore, as an argument for GPU's action in attempting to restrict the license.

The symbolic power of the GPU attempt at restricting the GPL licenses is not, however, a strong argument for changing clause six. First, the existence of clause six did not preclude the GPU action; arguably, the existence of clause six magnified the power of GPU's act. It was only after the conflict with FSF and OSI was made public that the GPU's symbolic act became well known. Without that conflict, it is not clear that the symbolic act would have had much effect; indeed, it had little visible external effect for almost a year.

Second, GPU might have had similar or even more symbolic power if it had made a public act of abandoning SourceForge.net and its GPL compliance because of the possible military use of their application. This act of conscience would have inflicted significant costs on the GPU project and would perhaps have intensified the ensuing debate. GPU's "little taunt" might have been a more significant act if it included the aspect of self sacrifice. This more powerful statement and sacrificial act could have been interpreted more as a statement directed against military uses of software and less as a controversy about GPL licensing. That is, GPU could have acknowledged the usefulness of FLOSS and regretted abandoning it because of a principled stance against military uses of their particular application. Instead, their reversion to the GPL after being challenged reduced the symbolic power of their act.

## Conclusions

The GPU no military use clause dramatized important issues for all software developers. It also brought into clear focus the tradeoff between allowing unfettered access to source code and the ability to control the use of a programmer's creative effort. The eventual retraction of the GPU restriction illustrates the power FSF and OSI now have in the OSS movement. The case illustrates the centrality of the FSF freedom 0 in the philosophical underpinnings of FLOSS, and some of the consequences of supporting that freedom consistently. The case and the controversy surrounding the case is also a demonstration that the transparency of FLOSS (as contrasted with the more private nature of proprietary software) encourages widespread discussion and lively debate about professional ethics issues in software development.

### Acknowledgments

The author acknowledges Don Gotterbarn for being a leader in computer ethics who recognizes crucial issues clearly and quickly, and for his outspoken advocacy of professional ethics for software developers. The author also acknowledges Fran Grodzinsky and Marty Wolf for their important explanations of the ethics of open source software.

### References

Asimov, Isaac. 1950. *I, Robot*. New York: Doubleday & Company.

Stallman, Richard. "The GNU project." http://www.gnu.org/gnu/thegnuproject.html.

Stallman, Richard. "Copyleft: pragmatic idealism." http://www.fsf.org/licensing/essays/pragmatic.html.

FSF. "Free software definition." http://www.gnu.org/philosophy/free-sw.html.

"Links to other free software sites." http://www.gnu.org/links/links.html#FreedomOrganizations.

FSF. "What is copyleft?" 2006. http://www.gnu.org/copyleft/copyleft.html.

Garner, W. David. "SHARE, IBM user group, to celebrate 50th anniversary," *Information Week* (Aug. 17, 2005). http://www.informationweek.com/software/soa/169400167.

Gasperson, Tina. "Open source project adds 'no military use' clause to the GPL." *NewsForge* (August 14, 2006). http://gpu.sourceforge.net/.

SourceForge.net. "GPU, A Global Processing Unit." http://gpu.sourceforge.net/.

Grodzinsky, F., K. Miller and M.Wolfe. "Ethical Issues in Open Source Software," *Journal of Information, Communication and Ethics in Society* 1 (2003): 193-205.

Ethical Perspectives. "Free software or proprietary software? Intellectual property and freedom." *Journal of the European Ethics Network*. http://www.ethical-perspectives.be/page.php?LAN=E&FILE=subject&ID=364&PAGE=1.

Klepas.org. Blog. "Military use a 'freedom?'" (April 17, 2006). http://klepas.org/2006/08/17/military-use-a-freedom/.

League for Programming Freedom. "Software patents." 2006. http://lpf.ai.mit.edu/Patents/patents.html#Intro.

OSI. History: documents. http://www.opensource.org/docs/history.php.

OSI. Licensing. http://www.opensource.org/licenses/.

OSI. Open Source definition. http://www.opensource.org/docs/definition.php/.

OSI. Welcome. http://www.opensource.org/index.php.

Prasad, Ganesh. "Open source-onomics: examining some pseudo-economic arguments about Open Source," *Linux Today* (April 12, 2001). http://www.linuxtoday.com/infrastructure/2001041200620OPBZCY--.

SHARE. About us. http://www.share.org/About/.

Turner, David. FSF - Censorship envy and licensing (Feb. 11, 2005). http://www.fsf.org/blogs/licensing/20050211.html.

US Code: Title 42, 3604. 2006. Discrimination in the sale or rental of housing and other prohibited practices. http://www4.law.cornell.edu/uscode/html/uscode42/usc_sec_42_00003604----000-.html.

Wikipedia. Open-source software. http://en.wikipedia.org/wiki/Open_source_software.

Wolf, M., K. Bowyer, D. Gotterbarn, and K. Miller. Open source software: intellectual challenges to the status quo, *ACM SIGCSE Bulletin* 34 (2002).

Wolf, M., K. Bowyer, D. Gotterbarn, and K. Miller. b.Open source software: intellectual challenges to the status quo: Presentation slides. 2002. http://www.cstc.org/data/resources/254/wholething.pdf.

Yesh.com. No military use clause in GPL (August 14, 2006). http://www.yesh.com/blog/2006/08/14/no-military-use-clause-in-gpl/.

Zimmermann, P. Phil Zimmermann's homepage. http://www.philzimmermann.com/EN/background/index.html.

# FROM THE CHAIR

As I begin my last six months as chair, important changes are occurring within the PAC committee. Peter Boltuc (University of Illinois–Springfield) has been named co-editor of this *Newsletter* and will begin serving as an *ex officio* committee member. (Thanks to Peter for putting out this edition of the *Newsletter*.) As stated in my last report, Michael Byron (Kent State University) has begun serving as associate chair and will do so until July 1, 2007, when his term begins.

The 2006 Eastern division APA meeting has just concluded, and several important PAC committee events occurred there. Committee members continue to be active in planning sessions for APA conferences. Such sessions are one of the primary means by which the Committee carries out its charge of informing the profession concerning issues related to computer use. At this Eastern division meeting, the Committee awarded the Jon Barwise prize to Jim Moor (Dartmouth College). Jim's presentation ("The Next Fifty Years of AI: Future Scientific Research vs. Past Philosophical Criticisms") generated a lively discussion that continued during a subsequent reception. (Photos show Jim Moor and Jim with myself and committee member Amy White (Ohio University). Another session, organized by committee member Chris Grau (Florida International University), addressed the topic of "Robot Ethics" and included presentations by James Moor (Dartmouth College), J. Storrs Hall (Institute for Molecular Manufacturing), Michael Anderson (University of Hartford), and Susan Anderson (University of Connecticut–Stamford). Commentaries were provided by Selmer Bringsjord (Rensselaer Polytechnic Institute), Colin Allen (Indiana University), and Andrew Light (University of Washington).

Committee members also joined in the discussion during a session sponsored by the International Association for Computing and Philosophy (IACAP). During this session, ("Conflicts, Compromises, and Responsibility in Open Source vs. Proprietary Software Development"), presentations were made by Scott Dexter (CUNY–Brooklyn), Keith Miller (University of Illinois–Springfield), and John Snapper (Illinois Institute of Technology). I'm happy to report that a healthy interaction between the PAC committee and IACAP remains strong, as evidenced by the dynamic nature of this session.

On a more somber note, it is my task once again to note the passing of a good friend of the PAC committee. On September 18, 2006, Preston Covey of Carnegie Mellon University passed on due to adult complications from childhood polio. It is difficult to articulate the impact of Preston's dynamic personality and his invigoration of the interplay between philosophy and computing, and I will have more to say on this score in a subsequent article. For now, I'll note that Preston was both a leader and a cultivator in respect to this emerging field. Beyond helping to define the conceptual constitution of computing and philosophy, he helped to establish a community and to define the value of its work (often helping younger scholars to sense the importance of their own work). Through the activities of the Center for the Design of Educational Computing and the Center for the Advancement of Applied Ethics and Political Philosophy, and via multiple conferences held at Carnegie Mellon University, he provided a geographical and intellectual center for the CAP community. That community has prospered and expanded internationally and now embodies the International Association for Computing and Philosophy. Recent chairs of the PAC committee, including myself, Robert Cavalier, Terry Bynum, and Jim Moor have had the good fortune of being Preston's colleagues, and the Committee has greatly benefited from this association.

Looking to the future, the PAC committee will sponsor a session at the 2007 Central division meeting in April. Jerry Kapus (University of Wisconsin–Stout) will chair a session featuring Renée Smith (Coastal Carolina University, "Lectures and Discussions for the Virtual Classroom"), Scott Chattin (Southeastern Community College, "Designing Distance Philosophy Courses in a Community College Setting"), Peter Boltuc (University of Illinois–Springfield, "A Blended Argument"), and Marvin Croy (University of North Carolina–Charlotte), "Understanding the 'No Significant Difference Phenomenon'"). At that same conference, I will chair a session sponsored by IACAP, which features Helen Nissenbaum (School of Law, New York University, "Websearch Privacy in a Liberal Democracy: The Case of TrackMeNot") and Ned Woodhouse (Rensselaer Polytechnic Institute, "Toward a Political Philosophy of Information Technology"). Michael Kelly (University of North Carolina–Charlotte) will provide commentary.

That's all of now. Let me know if you have questions, concerns, or suggestions related to PAC committee activities, including ideas for Committee sessions at APA meetings.

Marvin Croy
University of North Carolina–Charlotte
mjcroy@email.uncc.edu

# NA-CAP@LOYOLA 2007

## The Annual North American Meeting of the International Association for Computing and Philosophy

### July 26th – 28th, 2007

### Call for Proposals

The 2007 North American Computing and Philosophy Conference will be held at Loyola University's Water Tower Campus in Chicago from July 26th to the 28th. The theme for this year's conference is Open Source Software and Open Access Publication. Keynote speakers are Richard Stallman, founder of the GNU project, and Peter Suber, a leader in the open access movement. (Visit the conference website at http://na-cap.osi.luc.edu.)

In addition to this theme, the conference will also include its usual array of topics in information ethics, cognitive science, AI, robotics, cultural and social issues, simulations and modeling, distance learning, computational logic and linguistics, and electronic and teaching resources. (To get a feel for CAP conferences in general and the range of presentation topics typical of one of our conferences, visit http://ia-cap.org.)

Currently, the program committee is soliciting proposals for the conference. Please submit electronically an extended abstract of approximately 1,000 words, targeted to one of the topic areas below. Include your name, institutional affiliation (if you have one), email and snail mail addresses, a title for your presentation, and a short, 2-3 paragraph abstract for use on the conference web site. Attach supplemental materials (such as links, PowerPoint slides, etc.) as you see fit.

Send two copies of your proposal, one to the committee member for the appropriate topic area below and the other to Anthony Beavers, program chair of the conference, at afbeavers@gmail.com.

DEADLINE: March 1st, 2007

### The 2007 NA-CAP Program Committee:

**Information and Computer Ethics**
Terry Bynum (bynumt2@southernct.edu)
**Cognitive Science, AI, and Robotics**
Selmer Bringsjord (selmer@rpi.edu)
**Social, Cultural, and Metaphysical Issues**
Charles Ess (cmess@drury.edu)
**Simulations and Computational Modeling**
Branden Fitelson (branden@fitelson.org)
**Issues in Distance Learning**
Peter Boltuc (pbolt1@uis.edu)
**Computational Logic and Linguistics**
Patrick Grim (pgrim@notes.cc.sunysb.edu)
**Electronic Scholarly Resources**
Anthony Beavers (tb2ue@aol.com)
**Electronic Teaching Resources**
Michael Byron (mbyron@kent.edu)
**Student Track - Grads and Undergrads**
Matt Butcher (mbutche@luc.edu)