APA Newsletters
NEWSLETTER ON PHILOSOPHY AND COMPUTERS

Volume 11, Number 1 Fall 2011

apa
The American Philosophical Association
University of Delaware
Newark, DE 19716

 ISSN 2155-9708

FRANK MCCLUSKEY
"Reflections from Teaching Philosophy Online"

TERRY WELDIN-FRISCH
"A Comparison of Four Distance Education Models"

KRISTEN ZBIKOWSKI
"An Invitation for Reflection: Teaching Philosophy Online"

THOMAS URBAN
"Distance Learning and Philosophy: The Term-Length Challenge"

FEDERICO GOBBO
"The Heritage of Gaetano Aurelio Lanzarone"

APA Newsletter on

Philosophy and Computers

Piotr Bołtuć, Editor | Fall 2011 | Volume 11, Number 1

From the Editor

Peter Boltuc
University of Illinois at Springfield

I used to share the general enthusiasm about web-only publications, especially if they are open access. Not so much anymore. There used to be a general understanding that once content of any value is put on the web it never dies away, always mirrored by numerous open access websites. Yet, when the APA website went through the much needed upgrades and the newsletter went down for the last Summer and Fall, only one issue (and a few random articles) were available online. At the time when a number of "traditional" journals are pushed by publishers to go entirely online the experience seems sobering. At some point in time, when the hosts of such journals go belly up, or lose interest in a given area, those publications may disappear entirely. Not an encouraging prospect that runs counter to the predictions of informational cornucopia shared with our readers a few issues ago by Luciano Floridi. Talking about a cornucopia, the term seems like the best characterization of events that accompanied the award of Doctorate Honoris Causa of University of Suceava to Luciano this October. It is rare for a philosopher of information to receive such an honor; congratulations are in order.

I am quite sure that the article featured in this issue, in which Jaakko Hintikka presents logic as a theory of computability, and computation as a logical proof, shall be reprinted and quoted many times due to its potentially seminal implications. The note from Dan Kolak presents Hintikka's article in more detail. The article by Hector Zenil is aimed to demonstrate that algorithmic information should be treated far more seriously by philosophers. Darren Abramson and Lee Pike, in their interesting article, argue for a rather high safety standard of using *formal methods* to check programming in many ethically important areas. Pentti Haikonen responds to Shanahan's commentary on his earlier paper in this *Newsletter*. Pentti argues not only that human consciousness is less unified than we used to think but also that cognitive architectures should be developed likewise.

The block of papers devoted to online learning in philosophy is a follow-up on a section organized by our committee in Minneapolis at the APA's Central Division meeting, though as a matter of fact, of the papers included in this issue only the one by Kristen Zbikowski was actually presented there. Ron Barnette, one of the original active members of this committee, opens the block with some memories of his early work in online education and that of others. Frank McCluskey, one of the leading proponents of online learning nationally, looks back at his history of online learning in philosophy. Terry Weldin-Frish, in his informative paper, compares the experiences he had with online learning in philosophy first as a graduate student, reaching a Ph.D. entirely online, and later as a faculty member, at four different educational institutons in the UK and the US. Kristen Zbikowski presents a spirited defense of teaching philosophy online based on her experiences as an online student and then a faculty member also teaching online courses. Thomas Urban raises a specific problem of the length of viable online philosophy courses.

At the last moment we add an obituary of Elio Lanzarone, a recent contributor to this *Newsletter* and a friend who passed away this fall.

From the Chair

Daniel Kolak
William Paterson University

I am pleased to announce that we have awarded the 2010 Barwise Prize to Jaakko Hintikka (Boston University) for his unique contributions to the advancement of logic, especially epistemic logic and IF (independence friendly) logic, which model not only parallel processing but the connections between computation and cognition, and which have had a direct impact on computer science and related applications.

Hintikka presented his Barwise talk at the Pacific Division meeting on Saturday, April 23, 2011, 4-6:00 p.m. The paper, provocatively entitled "Is Computer Science a Branch of Logic?" has proven interesting to say the least.

Modern logic is often called "symbolic logic" because it was supposed to be a realization of Leibniz's vision of reasoning—at least deductive reasoning—as calculation with symbols. In his talk, Hintikka raises the reverse question that we now can pose in precise terms, thanks to the advancement of logic and computability theory: Can all computation be considered as a logical deduction? If the logic used is our common first-order logic (FOL), the answer is *no.* Hintikka has succeeded in diagnosing the reason for this failure. It turns out to be a flaw in our received first-order logic that goes back to Frege. Fortunately this flaw has been corrected in the improved version of first-order logic that Hintikka has himself developed. This new basic logic is known as IF (independence friendly) logic. By its means, each computation can after all be interpreted as a logical deduction.

This latest result of Hintikka's has major philosophical implications. Philosophers and laypersons have asked: Does a computer think? Now we know that if it does, it uses the same logic as we do.

Hintikka's result opens also new constructive possibilities. It puts the entire logical theory to the service of computer science.

As an illustration of these possibilities, Hintikka sketches in intuitive terms an outline of a solution to the well-known P vs. NP problem. If his argument holds, it constitutes an answer to one of the famous Millennium Prize problems.

Special Session

Special Session on "Machine Consciousness"

A Note from David L. Anderson

(Sponsored by the APA Committee on Computers and Philosophy)

February 17, 3:00-6:00 pm

Ned Block, one of the world's most influential researchers on consciousness, will give the opening paper of this session that will explore fundamental issues concerning the nature of consciousness, through the lens provided by a focus on "machine consciousness." Peter Boltuc will take Block's work as his point of departure as he explores a non-reductive approach to machine consciousness. Terry Horgan, vigorous defender of "phenomenal intentionality," will bring that perspective to Searle's "Chinese Room Argument." And finally, Robert Van Gulick will bring to the table his critical analysis of the subject, which (we are guessing) will result in a challenge to those who want "phenomenal consciousness" to do too much of the heavy lifting. Please join us.

Chair: David Anderson

3:00 Ned Block, "Can thinking about machines help us understand consciousness?"

3:35 Peter Boltuc, "Non-reductive machine consciousness"

4:10 Terry Horgan, "The Real Moral of the Chinese Room: Understanding Requires Understanding-Phenomenology"

4:45 Robert Van Gulick, "Humans and other conscious machines — one way or many?"

5:20 Open Discussion

5:50 "Parting Shots" — each speaker has 2 minutes for a final comment

Featured Article

Logic as a Theory of Computability

Jaakko Hintikka
Boston University

1. Different frameworks for a theory of computability

A general theory of computability must rely on some conceptual framework or other in which the different steps of computation can be codified and by reference to which a theory of computability can be formulated. (For this theory, see, e.g., Davis 1958 or Clote and Krajiček 1993.) Different frameworks bring different resources to bear on the structure of computations. For instance, the Turing machine framework relates the general theory of computation to questions of computer architecture. The lambda calculus framework has facilitated the development of the denotational semantics for computer languages. The present-day theory of computability can be said to have come about when the main types of framework were shown to yield the same concept of computability.

One frequently used framework is recursion theory. (See here, e.g., Rogers 1967 or Phillips 1992.) In it the notion of a computable function is captured by means of generalization of the familiar use of recursive definitions in elementary arithmetic. In this framework, computable functions are defined as (general) recursive functions. (In the following "recursive function" means what is sometimes called general recursive function.) The class of recursive functions is defined as the smallest class of functions containing zero, successor, and projection functions and closed with respect to the operations of composition, primitive recursion, and minimization. Here a projection function P^n_m takes us from the n-tuple $x_1, x_2, ..., x_m, ... x_n$ to x_m. The minimization function f_g associated with a two-argument function g(x,y) has as its value $f_g(y)$ the smallest x such that g(x,y) = 0, with the understanding that for all values of $z < f_g(y)$, g(z,y) is computable but $\neq 0$. The possibility of forming functions by composition or by projection is built into the usual notation of logic. Recursive functions can be either total or partial, that is defined only on a subset of natural numbers.

As a preparation for the treatment of minimization functions it is useful to introduce explicitly the usual notion of minimum through the following recursive equations:

(1.1) min (x,0) = 0
min (s(x), s(y)) = s(min(x,y))
min (x,y) = min (y,x)

Here s(x) is the successor of x. As with other recursive definitions, the value min (x,y) is determined by the values min for smaller arguments, and hence can be computed in the finite number of steps. Then x ≥ can be defined as the equation min (x,y) = y.

2. Computations as logical proofs

There is one more class of formal operations of which you can ask whether they can serve as a framework of computability theory. They are formalized deductions. The basic idea of trying to use them as a framework for computations is clear. It is to construe the computation of the value a = f(b) of the function f for the argument b as a deduction of the equation a=f(b) in a suitable system of number theory.

At first sight, this looks easy. For instance, computations of value of a primitive recursive function from their defining equations is accomplished by the repeated use of two rules

(S.1) Substitutivity of identicals

(S.2) Substitution of a term for a variable

We can restrict (S.2) to substitutions of constant terms for variables. It is convenient to think of all the applications of these rules as taking place in a conjunction of equations whose variable are all bound to suppressed outside universal quantifiers. But what are the relevant equations? What other premises are perhaps needed? (They can be thought of as additional conjuncts.)

These equations and other premises should express operations (listed above) used to form general recursive functions. All these operations can obviously be expressed in the form of equations except for the formation of minimization functions. In order to accommodate these we can extend the class of deductive premises used in the deductions that are interpretable as computations. This can be done by introducing as the relevant extra assumption the conditionals

(2.1) $(g(x,y) = 0) \supset (g(f_g(y),y) = 0)$
$(g(x),y) = 0) \supset \min((x,f_g(y)) = f_g(y))$

How can computations using (2.1) be captured deductively? These deductions cannot any longer be merely sequences of equations. They must also involve applications of some propositional rules, since we are now dealing with propositional combinations of equations. Negations of formulas are not involved, for negations occur in (2.1) only in front of identities.

Very little reasoning beyond (S.1)-(S.2) is needed. If for some value b of y it is the case that

(2.2) $(\forall x)(g(x,b)\neq 0)$

In this case a proof for (2.2) can be found in a finite number of steps, since g is assumed to be recursive. For the same reason, we can compute, for a given value b of y, the values of g(0,b), g(1,b), ... If (2.2) is not true, then for some d we have g(d,b)=0. Then $f_g(b)$ is the smallest x for which g(x,b)=0. This will be found by computing g(d-1, b), g(d-2,b), ... This presupposes that g(0,y), g(1,y),...,g(d,y) are defined; otherwise $f_g(y)$ is not defined. But this is in keeping with the usual recursion theory. (See Phillips 1992, p.112.)

The only deductive rules needed for this purpose are (S.1)-(S.2) plus suitable propositional inference rules. This argument shows that if computations using a function g(x,y) can be construed as first order deductions, then so can computations using the corresponding minimization function $f_g(y)$.

Thus all computations of the values of a general recursive function can be construed as deductions. The premises of these deductions include their defining equations plus definitions of the minimization functions used in the computation. These definitions can be taken to be of the form (2.1).

The recursive function thus defined is usually a partial function. Only when it is the case that

(2.3) $(\forall y)(\exists x)(g(x,y)=0)$

do we have a total function.

If for some reason an instance of (2.3) is false for some value b of y, then a proof of the negation

(2.4) $(\forall x)(g(x,b)\neq 0)$

can be found in a finite number of steps.

3. Deductive proofs as computations

This does not answer the question whether logic can serve as a theoretical framework for computation theory. For this purpose it has to be required that sets of premises of the deductions that reproduce any given computations can somehow be represented by logical formulas doing the same job.

At first sight, this seems easy. We can transform first-order formulas into premises for a computation on the basis of equations as follows:

(a) All formulas are transformed into a negation normal form.

(b) All predicates are replaced by their characteristic functions.

(c) All existential quantifiers are eliminated in terms of their Skolem functions. That is, each subformula of the form $(\exists x)F[x]$ is replaced (in context) by

(3.1) $F[f(y_1, y_2,...,t_1, t_2, ...)]$

where (Q_1y_1), (Q_2y_2), ... are all the universal quantifiers (in a negation normal form) within the scope of which $(\exists x)$ depends and t_1, t_2, ... are the terms on which $(\exists x)$ depends. (Note that the variables y_1, y_2, ... may occur in t_1, t_2, ...). The function f is the Skolem function correlated with the occurrence of $(\exists x)$ in question.

This elimination of existential quantifiers is assumed to be carried by step inside out.

(d) Universal quantifiers are moved to the beginning of the formula (or conjunction of formulas in question. They can be thought of as being suppressed.

The result is a propositional combination of equations and negations of equations that can be used for computing values of functions as indicated earlier.

Obviously, in this way we can only compute Skolem functions with the help of other Skolem functions.

4. Restrictions on Skolem functions

This nevertheless implies a restriction on the functions that can be computed in this way. Not any set of equations that can be used for this purpose can be obtained as Skolem functions of formulas of traditional first-order logic (or of their conjunctions). This is because of the (labelled) tree structure of traditional first-order formulas. Because of this structure, the argument set $(y_1,y_2,...,t_{y1} ,t_2,y)$ of the Skolem function f in (3 λ) is the set of variables bound to universal quantifiers with the scope of which $(\exists x)$ occurs (plus certain constants). Because of the tree structure (nesting structure) of quantifiers, the members of these argument sets may be function terms. Each formally different function term (including as special cases variables and constant) is considered a different argument. Since the (initial) universal quantifiers do not depend on anything, the argument sets containing only them do not matter.

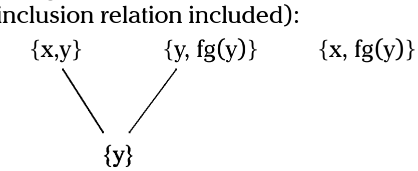
Because of the tree structure (nesting of scopes) these argument sets of Skolem functions must have the same tree structure. They must be partially ordered by class inclusion with all chains linearly ordered (not branching) in the upwards direction.

Not all sets of equations (and propositional contributions of equations) that can be used for computing recursive functions exhibit such a structure (in the argument sets of their functions). In other words, the partial ordering in question must not exhibit any branching upwards. In such cases, we are dealing with a set of equations (and their propositional combinations) that can be used in computation but cannot be interpreted as a set of deductive premises for a corresponding deduction.

The functions whose argument sets are ordered in this way cannot be Skolem functions of any set of ordinary first-order formulas. Hence, not all the sets of defining equation of general recursive functions can be obtained from ordinary first-order formulas.

For instance, if the assumptions that include (2.1) plus the defining equations listed in the sec. 1, the structure of the relevant argument sets of the relevant functions is the following (with inclusion relation included):

(4.1)



Here the upwards chain from {y)} branches. The relevant set of defining equations cannot be obtained as Skolem functions of a formula in ordinary first-order logic.

However, if (2.3) is true we can drop the antecedent g(x,y)=0 from (2.1). Then the upwards branching disappears from (4.1). This is the case when the definition (2.1) yields a total function. Accordingly, all total functions are computable by deductions in ordinary first-order logic.

5. Limitations of the received first-order logic

Hence it follows that the received logic (the usual first-order logic) cannot serve as a framework for a general theory of computation. However, this is merely due to certain unnecessary limitations of this logic which go back all the way from its first foundation by Frege. Frege overlooked part of the semantical job of quantifiers. This job does not involve only ranging over a class of values and expressint the nonemptyness or exceptonlessness of certain (usually complex) predicates in this class. By their formal dependence on each other quantifiers also express the actual dependence of their respective variables on each other.

In the received first-order logic these formal dependencies between quantifiers is expressed by the nesting of their scopes. In this way we can only express patterns of dependencies that exhibit a tree structure of the kind mentioned above. It is obvious that this restricts the kinds of set of functions that can be represented in ordinary first-order logic. What has been found here is an example of these restrictions.

These restrictions are partly removed in what is known as independence-friendly (IF) logic. It is obtained from the received first-order logic by allowing an existential quantifier (∃y) occurring (in a negation normal form) within the scope of (∀x). This can be expressed by writing it as (∃y/∀x). The same effect could be reached without any new symbols by liberating the use of parentheses.

In IF logic the law of excluded middle does not automatically hold. Hence it is well suited for discussing partial functions, for instance general recursive functions. Accordingly, we have to distinguish the strong (dual) negation ~ from the contradictory negation ¬. The use of the letter, albeit only sentence-initially, characterizes what is known as extended IF logic (EIF logic). This logic has two halves. The ∑-half is unextended IF logic while the ∏–half consists of contradictory negations of IF formulas.

The use of IF logic as a framework of formal reasoning (formal computation) may seem inappropriate in that there is now complete axiomatization (no complete set of rules of inference) for it. However, in recursion theory we can restrict ourselves to formulas in which all negations (and sequences of negations) of any kind form atomic formulas or identities. For this fragment the usual rules of first-order logic (other than rules for negation) are valid and can be used in deductions.

Another aspect of the same situation is that while IF logic does not admit of a complete proof procedure it has a complete disproof procedure. (A cut-free proof procedure for traditional first-order logic can serve as one.) Conversely, the $\prod^1_1$– fragment of EIF has a complete proof procedure but not a complete disproof procedure.

The same semantics works for an extended independence-friendly (EFI) logic to which the contradictory negation ¬ is admitted, if only into sentence-initial positions.

6. EIF Deductions and Computations

It is now possible to show how one can transform any given computation of the value of a recursive function for a given argument into a logical deduction. A way to do so is obvious to a competent logician, but a survey of what is involved is nevertheless in order. The logical operations take place within the scope of a string of initial universal quantifiers. In that scope, we have a conjunction whose conjuncts specify the operations used in forming the function in question. They include:

(i) The recursion equations for the primitive recursive functions used for the purpose

(ii) The instances of the projection operation used, e.g., $P^1_2(g(x),h(x,y)) = g(x)$

(iii) Defining equations for all the intermediate functions used.

(iv) For each of the minimization functions used, for instance $f_g(y)$ obtained from g(x,y), the conjuncts will include (2.1)

The computation proceeds by means of (S.1)-(S.2). A branching structure is created by the disjunctions in the way indicated in sec. 2 above.

The task is to transform this computation into a deductive argument. Since all the rules used in the computation are in effect valid deductive steps, what has to be done is to replace functions by predicates and show how the computational line of reasoning can be extended so as to be carried out in terms of the resulting formulas.

For the purpose, we do the following:

(a) For each different term occurring in the equations we introduce a variable (a constant), if it was not already. Then we introduce to the conjunction the "definitions" of all the new variables as simple functions of others. After this change, all (function) terms are simple (i.e., involve no nesting of functions).

An example makes these explanations clear. For instance, the recursion equations for addition a(x,y) are

(6.1) a(0,y) = y

(6.2) a(s(x),y) = s(a(x,y))

Here s(z) is the successor of z. These equations contain the terms 0,y, a(0,y), s(x) , a(s(x),y), a(x,y), s(a(x,y)). The new equations with their additional variables might be the following:

(6.3) z = s(x) v = a(z,y) u = a(x,y) w=s(u)

The original recursion equations now become

(6.4) a(0,y) =y a(z,y) = s(u)

In general terms, after this change, the computation becomes a matter of applying (S.1)-(S.2) to the new equations. For instance, the computation of a(2,1) is now accomplished by the following substitutions where we take into account the definitions 1 = s(1), 2=s(1), 3=(s(2):

(6.5) a(2,1) = a(s(1),1) = s(a(1,1)) = s(a(s(0),1)) = s(s(s(0)) = s(s(0)) = s(2) =3

Each identity results from one of the equations (6.3)-(6.4) by a substitution of constants for variables.

(b) For each function, say g(x,y), we introduce a corresponding predicate G(x,y,z) and add to the conjunction two formulas

(6.6) (∃z//∀x,∀y) G(x,y,z)

(6.7) (G(x,y,w) & G(x,y,u)) ⊃ (w=u)

Here (∀x)(∀y)(∀w)(∀u) are among the initial quantifiers. Notice that if we did not have the slash notation available, we could not make sure (by means of a linear ordering of different quantifiers) that the value of z in (6.6) depends only on x and y. This is where the impossibility of construing computations as deductions in the received first-order logic shows up.

(c) Each atomic formula or identity A(g(x,y)) containing the term g(x,y) is replaced by

(6.8) (∀z)(G(x,y,z)⊃A(z))

This can of course be done to all the terms in A at the same time. The quantifier (∀z) is thought of as being moved to a sentence-initial position.

These steps eliminate all functions in terms of predicates.

(d) This change correlates with each simple term, say g(x,y), an existential quantifier (∃z) that depends only on (∀x), (∀y). The substitution of numerical values for x and y combined with an existential instantiation with respect to z yields a constant that is different for each different term g(x,y).

When all these changes have been made, the computation of the value of a recursive function is transformed into a formal logical proof. In the proof, each application of (S.1) remains a substitution of identicals. All substitutions needed are substitutions of constant terms (terms without variables) for variables. Each constant term is in the logical proof introduced by an application of existential instantiation to a formula (6.6). It follows that the number of different terms introduced in the computation equals the number of introductions of new constants in the proof by existential instantiation.

There is another, theoretically simpler way of turning computations into formal deductions. Assume that a function of f can be computed from a conjunction of equations E (perhaps conditional ones like (2.1)) involving the successor function S and a number of auxiliary functions $g_1, g_2, \dots, g_n$. Then obviously the computations can be seen as a deduction of equations of the form

(6.9) $f(a) = s(s(\dots (s(0))$

from

(6.10) $(\exists g_1)(\exists g_2) \dots (\exists g_n) E$

But (6.10) is a $\sum^1_1$ sentence and the deduction can therefore be transformed into a formal deduction in extended IF logic, indeed in its $\prod^1_1$ half for which there exists a complete (formal) axiomatization (proof procedure).

7. IF logic as a framework of computation

Thus EIF logic can serve as a general framework of computation. Computations can be transformed into EIT deductions, and deductions can be replaced by computations. Questions concerning computation can become problems concerning the deductive structures (the structures of logical consequence relations) in IF or EIF logic. (IF logic is equivalent to the $\sum^1_1$ fragment of second-order logic while EIF logic also contains as a mirror image of IF logic an equivalent of the $\prod^1_1$ fragment.) For instance, different algorithms which in the simple equational calculus are codified in the systems S of equations correspond to different deductive premises. For instance, the recipe for forming the minimization function $f_g(y)$ can be expressed in IF logic in the form of an explicit premises:

(7.1) $(\forall y)\ (\exists x)(g(x,y)=0) \supset (g(f_g(y), y)=0))$
$(\forall y)(\forall x)((g(x,y)=0) \supset (\min(x,f_g(y)) = f_g(y))$

Hence questions of consistency and computational power can hence be studied by examining the deductive relations between IF formulas. Likewise, questions of the complexity of computational processes are translated into questions of the complexity of formal deductions. (For such questions, see, e.g., Li and Vitányi 1993.)

It can also be seen that if for a given value of y it is true that

(7.2) $(\exists x)(g(x,y)=0)$

then the difference between ¬ and ~ (2.1) does not make any difference. In this case, IF logic as distinguished from the received first-order logic is not needed. If this is the case for all values of y, then the computation of the values of $f_g(y)$ can be carried out in the ordinary first-order logic which hence is adequate as a framework for studying total recursive functions, as is indeed well known.

There nevertheless seems to be a major limitation of what can be done in this way. For there is no complete axiomatization (proof procedure) for IF logic. However, this is not an actual obstacle, for (as was noted) there exists a complete disproof procedure for IF logic, that is, a recursive enumeration of inconsistent formulas. (This follows from the compactness of IF logic.) In fact, a suitable cut-free proof procedure, such as the *tableau* method for the received first-order logic, or the tree method can serve this purpose.

Then how do you use the new framework to compute values of functions? A system of equations is coherent in an interesting sense if the following is valid in it for each function f:

(7.3) $(\forall y)(\exists x)(f(y)=x)$

Then for each numeral a you can disprove

(7.4) $\sim(\exists x)(f(a)=x)$

But you can disprove this only by deriving a formula of the form

(7.5) $\neg \sim(f(a)=t)$

for some constant term t. But all such constants are built from 0 by means of primitive recursive functions. Hence t can be taken to be a numeral.

Now f(a)=t must be either true or indefinite. In the second case, there cannot exist a different numeral n* such that f(a)=n*. For since f(a) can have only one value, it cannot have a definite value different from n. This is because if it had, the truth value of f(a) could not be indefinite either.

The detailed development of a logic-based theory of computability is too large a task to be undertaken in a single paper. The purpose of this paper is merely to clean obstacles from the way of such theory.

References

Clote, Peter, and Jan Krajiček, eds. *Arithmetic, Proof Theory, and Computational Complexity*. Oxford: Clarendon Press, 1993.

Cook, Stephen A. "The Complexity of Theorem-proving Procedures." *Proceedings of the Third Annual ACM Conference on Theory of Computing*, 1971, 151-58.

Cook, Stephen A. "The P versus NP Problem." In *The Millenium Prize Problems*, edited by J. Carlson, A. Jaffe and A. Wiles. American Mathematical Society, Providence, R.I., 2006, 87-104.

Davis, Martin. *Computability and Unsolvability*. New York: Dover Publications, 1958.

Fagin, Ronald. "Generalized First-order Spectra and Polynomial-time Recognizable Sets." *Complexity of Computation, SIAM-AMS Proceedings* 7 (1974): 43-73.

Hintikka, Jaakko, forthcoming, "On Skolem Functions in Proof Theory."

Hintikka, Jaakko. *The Principles of Mathematics Revisited.* Cambridge: Cambridge University Press, 1996.

Hintikka, Jaakko. "Distributive Normal Forms in First-order Logic." In *Formal Systems and Recursive Functions*, edited by J.N. Crossley and M.A.L. Dummett. Amsterdam: North-Holland Publishing Company, 1965, 47-90.

Hintikka, Jaakko. *Distributive Normal Forms in the Calculus of Predicates* (*Acta Philosophica Fennica* vol. 6). Helsinki: Societas Philosophica, 1953.

Hintikka, Jaakko, and Gabriel Sandu. "Game-theoretical Semantics." In *Handbook of Logic and Language*, edited by J. van Benthem and Alice ter Meulen. Amsterdam: Elsevier, 1997, 361-410.

Hintikka, Jaakko, and Gabriel Sandu. "What Is the Logic of Parallel Processing?" *International Journal of Foundations of Computer Science*, 6 (1995): 27-49.

Li, Ming, and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* New York: Springer, 1993.

Phillips, I.C.C. "Recursion Theory." In *Handbook of Logic in Computer Science*, vol. 1, edited by S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum. Oxford: Clarendon Press, 1992, 79-187.

Rogers, Hartley Jr. *Theory of Recursive Functions and Effective Computability*. New York: McGraw-Hill, 1967.

Sandu, Gabriel, and Merlije Sevenster. *Games and Logic: A Strategic Account of IF Logic*, Cambridge: Cambridge University Press, 2011.

Smullyan, Raymond. *First-Order Logic*. New York: Springer, 1963.

ARTICLES

When Formal Systems Kill: Computer Ethics and Formal Methods

Darren Abramson
Dalhousie University, da@dal.ca

Lee Pike
Galois, Inc., leepike@galois.com

> "Beware of bugs in the above code; I have only proved it correct, not tried it."
>
> —Donald Knuth

> "Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence."
>
> —Edsger Dijkstra

Abstract

Computers are different from all other artifacts in that they are automatic formal systems. Since computers are automatic formal systems, techniques called formal methods can be used to help ensure their safety. First, we call upon practitioners of computer ethics to deliberate over when the application of formal methods to computing systems is a moral obligation. To support this deliberation, we provide a primer of the subfield of computer science called formal methods for non-specialists. Second, we give a few arguments in favor of bringing discussions of formal methods into the fold of computer ethics.

1 Introduction

1.1 Our Goals

Consider the following general ethical question: "What moral obligations do software engineers and programmers have to apply formal methods to the fruits of their labor?" A subdiscipline of computer science called *formal methods* aims to dramatically increase the quality of software by mathematically *proving* programs correct as opposed to merely testing them.

This paper attempts to do two things. First, it is a call to arms to computer ethicists to answer this general question. Second, it argues that answering the question of how much formal methods is a natural and important part of computer ethics. We show that safety for computational artifacts is different in kind from conventional artifacts. In doing so, we provide a primer that will enable practitioners of computer ethics to better understand what formal verification is.

Some might argue,[1] in considering the first of our two goals, that it is trivially obvious that the question of how much expense should be taken to ensure the safety of software, especially when morally significant losses are at risk, is worthy of study by computer ethics. We are not merely making this claim. Indeed, this paper is motivated by the following pragmatic conundrum: Given that there are unique, expensive methods to ensure safety of computer systems, how come there is little philosophical discussion among computer ethicists regarding their use? We speculate the reason is a lack of widespread understanding that ensuring safety for computers is different from ensuring safety for other artifacts. The central purpose of this paper is *not dialectical but practical*: we hope to broaden the discipline of computer ethics, and by doing so, change social expectations of the safety of computer systems.

We take the view that doing applied ethics is not merely an intellectual pursuit, but also has as its end to change human practices. Just as bioethics demanded, and procured, considerations of autonomy in doctor/patient settings[2] we hope that computer ethics will demand, and procure, considerations of formal methods in the production and consumption of software.

1.2 Scope and Outline

In this paper, we advocate for the normative implications of the automatic formal system property of computers. To ground our investigation, we mostly restrict ourselves to *safety-critical* computer applications. These applications are ones in which failures may lead to the loss of human life. Such uses include aircraft control systems, automotive subsystems, computer systems for piloted spacecraft, and computerized medical devices. There is no reason, though, that in discussion of the question "how often should formal methods be applied to software?" computer ethicists must limit themselves to such applications.

In Section 3, we overview the field of formal methods, while in Section 4, we particularly focus on obstacles—both technological and cultural—for the field of formal methods, and we attempt to summarize the attitudes within computer science on the subject. Knowing what formal methods is, and explaining computer scientists' attitudes on its application, are prerequisites for an informed productive discussion of how often it should be applied.

Next we argue that consideration of formal methods belongs in computer ethics on general grounds of what justifies computer ethics. To more concretely motivate the place of formal methods within the discipline of computer ethics, in Section 5 we describe both how it informs and is informed by a debate central to computer ethics—the free software debate. Concluding remarks are given in Section 6.

2 Computers and Computer Ethics

2.1 Automatic Formal Systems

Before we explain what formal methods are, it is useful first to say something about what defines an object as a computer. It is the defining characteristic of computers that makes ensuring safety different for them than other objects. It is worth noting that this defining characteristic of computers has been well studied by cognitive scientists but not computer ethicists.

Cognitive scientists, by and large, agree that what makes computational systems distinct is that they are instances of, to use John Haugeland's term, *automatic formal systems* (AFS) (Haugeland 1989, Fodor 1990, Haugeland 1997, 8-9). An AFS is a concrete system that satisfies the following three properties:

1. *Token manipulation:* computers manipulate symbolic tokens according to formal rules (like games or logics).
2. *Digitalization:* computers have exact, repeatable results, as opposed to continuous systems (e.g., billiards or the weather).
3. *Finite "playability":* no computations take infinite time or require an "oracle."

Therefore, on this characterization, a computer realizes a formal system. A formal system can be described mathematically by a logic. Therefore, if we hold constant certain issues of fabrication and usage, we can, by applying a mathematical proof, *guarantee* that certain properties will hold of a computational system. Notice that we are only claiming a narrow implication: AFSes, implementing a sort of mathematical object, can in principle be subject to proof concerning certain abstract properties. This is not to say, of course, that there is a method such that for any abstract property and any formal system, the method can prove or disprove that the property holds of that system. To claim otherwise would be in violation of Church's Theorem, since assessing first order validity is an instance of determining abstract properties for a formal system.

Overall, the AFS property has received little attention in regards to its normative implications. Recent texts and anthologies in computer ethics instead focus on the unprecedented abilities of computers to store, transfer, and analyze information (Johnson 2001, Ermann and Shauf 2003, Winston and Edelbach 2006, Tavani 2007). We do not mean to dispute the factual claim that computers indeed have these unprecedented abilities that result in new ethical problems. Rather, we present the AFS property as being an additional property justifying the study of computer ethics, the thesis of which is central to the philosophical underpinnings of formal methods.

2.2 The Standard Justification

Let us consider AFS property with respect to a "standard justification" for computer ethics. In her influential textbook on computer ethics, Deborah Johnson argues, in answering the question, "Why computer ethics?" that the issues raised by computers are neither wholly new, nor wholly old.

> The ethical issues surrounding computer and information technology are not new in the sense that we have to create a new ethical theory or system. They call upon us to come to grips with new species. This means understanding the new issue in familiar moral terms, using traditionalist moral concepts. For the most part this is consistent with the traditionalist account because once connected to standard moral categories and concepts, the new issue can be addressed by extending familiar moral concepts to the new situation and drawing on our experience with other cases. However, the new species may not fit easily into standard categories and concepts: allowances for the special or new features of the new situation have to be taken into account. New species have special features and, as pointed out earlier, if we simply treat them as the same as other, familiar cases, we may fail to recognize how the new features change the situation in morally significant ways. (Johnson 2001, 22)

Johnson introduces what she calls the "genus-species account" of computer ethics to describe how we ought to address the ethical puzzles introduced by computing technology. It is dangerous, she argues, to think that we can understand our obligations by applying policies and arguments concerning older technologies and issues. Clearly, safety of manufactured products is a human concern with an existing history of substantial policies, laws, and arguments for responsibilities of various kinds. However, this genus of moral consideration, ensuring safety of human artifacts, cannot be applied to the species of AFSes in a traditional fashion.

Likewise, AFSes have a sort of complexity that is incommensurable with other artifacts (see Section 4.1). This new form of complexity is most obvious when we consider both the amount of testing needed to ensure that an artifact is safe and the certainty available from testing. On the one hand, testing tra- ditional artifacts can only yield probabilistic measures of safety. On the other hand, the verification of AFSes, with respect to their formal specification and under environmental assumptions, can yield absolute certainty of safety.

3 What Is *Formal Methods*?

Formal methods[3] is a subdiscipline of computer science. Formal methods takes the AFS property, described in Section 2.1, as an underpinning to its entire enterprise. Formal methods are mathematical techniques that are used to prove that particular programs (or, alternatively, hardware units) are *correct*—that is, that software or hardware satisfy a mathematical description of its desired functionality. This abstract description of desired functionality is called the *specification* of the hardware or software.

Formal methods, used to construct computer systems that behave as intended, can be contrasted with empirically testing the computing system. As an analogy, formal methods is related to conventional bug testing for computer design in much the same way that computational fluid dynamics (CFD) is related to wind tunnel testing for aircraft design (Rushby 1993).[4] The field of CFD is concerned with building mathematical models to investigate the aerodynamic properties of, say, an aircraft wing design or a boat's dynamics through water. CFD techniques allow dynamics to be analytically studied as opposed to building a physical model to test using a wind tunnel or other empirical experiment.

Similarly, manufacturers of computer systems employ engineers to be testers so that they can run their products through a "wind tunnel" (or "test harness," as it is called in software engineering). A test harness contains a large suite of inputs with the hope of covering both the full range of normal uses of the system as well as exceptional circumstances. As we have already described, through formal methods, one attempts to analytically analyze the system to prove properties about it for *any possible* inputs.

Broadly, three research directions are pursued in the field of formal methods:

1. *How to mathematically model formal systems and their environments.* Unlike other engineering artifacts that are mathematically modeled (bridge stresses, aircraft aerodynamics, etc.), many concepts in computer science are brand new, and a research challenge is simply how to mathematically model these concepts. Such concepts of course include modeling software and hardware and the environments in which they operate.

   Regarding environmental assumptions, any formal verification makes implicit and explicit assumptions. For example, in proving some property about a program, one might implicitly assume that the computer on which the software executes is not physically destroyed, that no bugs in the hardware change the intended program semantics, that the process thread executing the code is not terminated, and so on. Likewise, some assumptions might be explicit. For example, one might state as a hypothesis that an integer input to a program takes no more than a fixed number of bytes to represent it in binary. Other kinds of environmental assumptions are probabilistic. For example, fault-tolerant systems found in commercial aircraft are designed to mask a certain kind of and number of faults. The kinds of and number of faults is characterized by a *maximum fault assumption* (MFA). The MFA should hold with sufficiently high probability, but it is only probabilistic. However, supposing the MFA holds, the system should *provably* satisfy its specification. Thus, the correctness of the system is ultimately probabilistic even if its correctness under the MFA has been formally verified. *That formally verified system relies on possibly* *probabilistic hypotheses about the environment is a subtle point sometimes glossed over in the formal verification debate.*

In addition to modeling the environment, another challenge is to define and refine the abstraction of systems. For example, software models may be of the source-code semantics, an intermediate representation, a memory-aware model, or the machine code semantics. Hardware models may be of the microarchitecture, the register-transfer level, or of physical-layer protocols between components. System-level models may be of interacting subcomponents and interfaces at different levels of abstraction. Research is also devoted to modeling the connections between different levels of abstraction and aspects of systems and the software and hardware from which they are composed.

2. *How to mechanically check the correctness of mathematical proofs.* The output of this research is perhaps most visible in the development of *mechanical theorem-provers* (Wiedijk 2006). These software systems allow mathematics to be formulated[5] in a formal language, and the theorem-provers can check the correctness of proof scripts over the formulations that a user writes. Some theorem provers also provide a degree of automation to assist in the development of the proofs.
3. *How to automate mathematical proofs.* The focus of this research is on how to automatically generate proofs of correctness. The approach is limited by the state-explosion problem—even simple formal systems often contain an infeasible number of states to check whether some property holds of those states. However, techniques developed over the last two decades have made the approach feasible for systems with well over 10^{20} states (J.R. Burch et al. 1990). Decision procedures for decidable logics is also an active area of research, allowing infinite-state systems to be automatically verified (de Moura et al. 2004, Lahiri and Seshia 2004).

4 The History of the Debate

In Section 4.1, we discuss the apparent paradox of the mathematical modeling of formal systems lagging behind the success of mathematically modeling less abstract artifacts, such as bridges and aircraft, for example. We describe the status of ethical imperatives to use formal methods within computer science in Section 4.2. In Section 4.3, we very briefly describe the inroads that formal methods have made while focusing on what has motivated their use.

Taken together, these sections support the pragmatic purpose of this paper, by providing the computer ethicist with sufficient background on the use, benefits, and detriments of formal methods. We particularly try to convey a "computer scientists"' perspective, realizing we make some simplifications and generalizations.

4.1 Bridges, Planes, and Programs

On the face of things, bridges and mathematics do not appear to be intimately connected, but civil engineers and physicists have figured out how to mathematically model bridges to determine analytically all sorts of characteristics such as the maximum load a bridge can withstand, the effects of strong winds and earthquakes, and so on. The problem of mathematically modeling a bridge is essentially solved. The same story holds for studying the aerodynamics of aircraft, as mentioned in the previous section.

A priori then, if a computer is by definition an implementation of a formal, mathematical system, one may find it surprising that our ability to mathematically model programs is inchoate relative to bridges or planes. Why is that?

Three obstacles prevent formal methods from being widely adopted: formal requirements specification, the complexity of proofs, and the size of software systems.

4.1.1 Requirements Specification

Software requirements, particularly in safety-critical systems, are notoriously difficult to get right (Lutz 1993, Berry and Wing 1985). Indeed, the very idea of requirements being "right" may be incoherent. Requirements evolve as the needs and expectations of users evolve. Software gets used in ways that architects and developers would never have expected. Environmental interactions may produce unexpected results (e.g., a programmer for your web browser neglects to handle the case in which the user is navigating to a new web page and concurrently opens a new browser window, causing the application to crash).

While even formulating requirements is difficult, formalizing them is even more so. Stated informally, requirements often lack detail or omit corner-cases. For example, one might have the requirement that a hardware device multiply two numbers. Stating the requirement in a mathematical notation may lead the designers to consider the requirement in more detail: What happens if one of the numbers overfills a buffer? What if a buffer is modified during the computation? If the result overfills the result buffer, what should be returned? And so on.

On the other hand, formal methods may have its greatest payoff during the requirements engineering stage of a software project by forcing designers to overcome these difficulties early, before software has been implemented (Berry and Wing 1985).

4.1.2 Proof Complexity

Whereas the in CFD, a system is modeled with differential equations, programs are modeled using logic (propositional, first-order predicate, and second-order predicate logic are all used). A program can be modeled by determining the satisfiability of a logical formula modeling it—intuitively, a program is turned into a logical formula stating something of the form, "For all inputs, program P yields outputs with property X." The computational complexity of determining the satisfiability of even boolean formulas—the simplest of logics—is NP-complete, meaning it belongs to a mathematical class of "very hard" computational problems.

In mathematics, complexity is oftentimes managed by abstraction or problem-decomposition. While these techniques certainly pertain to software as well, their application is limited. Intuitively, the difficulty with decomposing software verification results from small local changes having global effects. For example, if a program changes a single "1" to a "0," the entire program could result in completely opposite behavior. De Millo, Lipton, and Perlis call this property the *discontinuity* property of software, as contrasted with the *continuous* functions reasoned about in Newtonian physics (Millo et al. 1979). That said, well-designed software is built to be compositional so that to the greatest extent possible, errors are localized. For example, a software bug that causes an application to crash should not cause the operating system to crash. In continuous domains, composition is inherent and depends less on good design principles. For example, a small, localized change to, say, the shape of an airfoil will have relatively small, localized effects on the aerodynamics of the aircraft.

In contrast, physical systems, like those analyzed using CFD models, are inherently continuous. CFD models can be used to model a system's behavior under nominal conditions.

To account for potentially anomalous behavior, a "safety factor" can be built in. For example, if an wing is expected to undergo x kilograms per square centimeter (kg/cm^2) under nominal conditions, it can be built to withstand $1.5x$ kg/cm^2, or some other safety factor. Due to the discontinuity of software, an analogous safety factor cannot be similarly computed.

4.1.3 Software System Size

The second reason that formal methods have not been more widely used is the size of software systems. For example, there is an estimated one billion lines of code on the new Airbus 380 airliner (not all of it is safety-critical, however) (Knight 2002a). Comparatively, the largest aircraft carriers in the world have on the order of one billion parts.

Together, the difficulty of mathematically modeling computers is more apparent—imagine if one missing bolt in an American aircraft carrier turned it into a Soviet submarine. That's software.

4.2 The "Computer Science Perspective"

Philosophers, logicians and computer scientists have written extensively on metaphysical underpinnings of formal methods and program verification (Barwise 1989, Smith 1985, James 1988); a nice annotated bibliography is available online (Rapaport 2007). Much of the debate has ranged over issues such as what it means to prove a program correct and what is the nature of a mathematical proof (i.e., is it sufficient for a machine to verify a proof or must a human do so?).

However, as far as the authors are aware, the normative implications of formal methods have largely been ignored by professional ethicists, and they have been mostly taken for granted amongst computer scientists. Of course, generalizing the viewpoint of all computer scientist professional or even all formal verification practitioners is impossible, but the normative questions rarely arise. When they do, there is a presupposition that they will further support program verification. The point might best be made by a quip heard by one of the authors at a recent formal verification conference: "It's true that no catastrophic commercial aircraft crash has been determined to be the result of faulty software, and that's a pity—program verification would be in greater demand if one had."

On the other hand, formal methods practitioners do not simply deliver diatribes against software and hardware developers. Formal verification practitioners are computer scientists themselves. They know that building correct software is difficult and that the most stringent program verification practices are not sufficiently mature to be used by non-experts in large-scale projects (indeed, skeptics often point out that few formal methods advocates verify programs they write themselves). Furthermore, industrial practitioners are employed by corporations producing the potentially faulty software. Consider that although an airbag designer may genuinely believe in the potential airbags to save lives, how many (publicly) condemn their automobile manufacturer for not installing airbags in every make and model? Likewise, no verificationist would seriously claim that *all* software should be verified.

Perhaps a fair characterization of the formal verification community's position is something along the following lines:

> Program verification is difficult. Our job is to figure out ways to reduce the difficulty so that the practice is feasible for industrial-scale endeavors. For security-critical and safety-critical systems, program verification is a "best practice" that should be employed, but we acknowledge the trade-offs between assurance of correctness and cost. Just as a car with anti-lock brakes is safer than one without, some individuals decide the trade-off of greater safety is not worth the additional cost. Our job is to make the practice more feasible and to advocate for formal methods, but only in a few circumstances would we claim that program verification is a moral imperative.

Let us expand the last point a bit regarding the moral obligation to practice program verification. Continuing our comparison to best practices in automobile manufacture, the infamous Ford Pinto case seemed to be a clear-cut moral issue. The Ford Motor Company calculated the cost of settling lawsuits due to a known unsafe design against the cost of recalling the vehicle (Dowie 1977). In comparison, malicious software development and deployment has not yet had similar high-profile cases of maliciousness. Indeed, faulty software has relatively rarely been deemed to be the result of gross incompetence, given the acknowledged complexity of software (see Section 4.3) and difficulty of formal verification. Indeed, we state the following *formal methods dilemma*:

> If formal methods is a best practice of software engineering, then an engineer who does not employ it is either negligent or incompetent. But formal methods is beyond the capability of typical software engineers (otherwise, why do we need formal methods experts and researchers?) or is too time-intensive to employ, so it cannot be considered to be a best practice today.

4.3 The Story So Far

Not surprisingly, advocates for formal methods are largely computer scientists, and even less surprisingly, the most ardent advocates are formal methods researchers and practitioners. The most significant inroads of formal methods into industrial design have come by way of economic motivation rather than ethical considerations. Perhaps the singularly most famous instance is Intel's "Pentium FDIV bug," which was a subtle hardware bug found in 1994 (Halfhill 1995). The bug eventually led to Intel's replacing defective chips, costing the company some half-billion dollars. Subsequently, Intel and other hardware companies began to augment their testing staff with formal methods experts.

In general, bugs have been less costly for software companies since software can be patched whereas hardware can only be replaced.[6] Nevertheless, more recently, software companies, such as Microsoft, also employ formal methods engineers to help improve code quality.[7] The motivation of a software company is increased reliability (e.g., reducing the likelihood of the "blue screen of death") and increased security.

Outside of the mainstream, safety-critical and security-critical computer developers have long advocated for—if not relied upon—formal verification (Neumann 1996). Safety-critical software, for example, includes automated flight-control software developed for commercial aircraft [Knight, 2002b]. Security-critical computers includes encryption devices [Pike et al., 2006]. Safety-critical and security-critical devices are usually designed to be as simple as possible with well-defined fixed functionality; see, for example, the L4 Microkernel[8] Project (Elphinstone et al. 2007). Their simplicity begins to make formal verification feasible.

Today, safety-critical and security-critical computer systems are becoming more pervasive. For example, next-generation automobiles may have "autopilot" functionality (Baleani et al. 2003). Software and robotics are used in medical devices (Jetley et al. 2006). Many of us rely on security protections of online banking, shopping, and so on. The pervasiveness is coupled with more complexity and increased functionality.

Amongst formal methods practitioners, there has been a conventional wisdom that lawsuits will soon be a prime motivator. The idea has been that liability lawsuits will be brought against hardware or software companies for losses (such as financial loss, security, life, etc.), and when it is shown that best practices in the field include formal methods, companies not employing them will be deemed negligent. The problem is, this has been the conventional wisdom for more than twenty years!

Such lawsuits have not materialized, despite estimates that faulty software and abandoned software development projects costing the U.S. economy at least $5 billion per year (Charette 2005). Why they have not—when, for example, extravagant liability lawsuits have been brought against companies in most other economic sectors—is an interesting question itself. However, bringing the discussion of formal methods to a wider audience might alter expectations by the public for the software they use.

5 Formal Methods and Intellectual Property

In this section, we further motivate the importance of considering formal methods within computer science. In Section 1, we began to motivate the ethical consideration of formal methods by describing the AFS property and examining it within the context of the "genus-species" justification for computer science. Here, we explore a topic that is squarely within the domain of computer ethics—free software—and argue that the consideration of formal methods informs the free software debate. First, we examine arguments in favor of the imperative that all software be free, in a very specific sense. We show that these arguments do not establish the desired conclusion. Then we argue that consideration of formal methods strengthens the position of free software advocates.

5.1 Stallman's Freedom Manifesto

To begin, let us briefly review the free sofwate debate. Computer ethics routinely treats the issue of whether intellectual ownership of software is ethically permissible, forbidden, or optional. In an influential series of documents, Richard M. Stallman has advocated the second position: he claims that allowing an individual or corporation to own software is unethical. A central thesis of Stallman's *GNU Manifesto*[9] is that free software is better justified than proprietary soft- ware (Stallman 1985). We will briefly explain what this means, how Stallman argues for this claim in very broad terms, and then describe the importance of formal methods for the issue of the freedom of software.

First, we must understand what Stallman means by free software. In an endnote to the *Manifesto*, he describes an ambiguity later resolved in another article, *The Free Software Definition*.[10] Stallman does not mean that individuals ought to have free access to physical instances of software (or, as he puts it in the definition, "You should think of free as in free speech, not as in free beer"). Rather, Stallman goes on to argue in the same article that individuals should have the following four freedoms with respect to software they acquire. We quote these freedoms below:

Freedom 0 The freedom to run the program, for any purpose.

Freedom 1 The freedom to study how the program works, and adapt it to your needs. Access to the source code is a precondition for this.

Freedom 2 The freedom to redistribute copies so you can help your neighbor.

Freedom 3 The freedom to improve the program, and release your improvements to the public, so that the whole community benefits. Notice that access to the source code is a precondition for this.

Stallman's *Manifesto* is directed at encouraging support for the creation of a fully featured operating system under these four freedoms. Moreover, he gives some arguments in favor that all software should be free in these senses. In justifying the GNU project, Stallman appears to argue that his idea for a software project follows from defensible, general principles for how software should be created.

For example, he writes, "GNU serves as an example to inspire and a banner to rally others to join us in sharing. This can give us a feeling of harmony which is impossible if we use software that is not free. For about half the programmers I talk to, this is an important happiness that money cannot replace" (Stallman 1985, 155).

Stallman also addresses independent considerations of utility. For example, consider the last two paragraphs of the same article: Stallman insinuates a utopian future based on sharing of information. Last, Stallman appeals to the historical justification used by Western societies for introducing protections on intellectual property. He claims first that copyright law has been introduced to benefit society, and not merely innovating individuals. Then, he writes,

> The fact that the easiest way to copy a program is from one neighbor to another, the fact that a program has both source code and object code which are distinct, and the fact that a program is used rather than read and enjoyed, combine to create a situation in which a person who enforces a copyright is harming society as a whole both materially and spiritually... (ibid., 159)

Let us assume, with Stallman, for the time being, that copyright law has only been, and only ever is, justified on the basis of benefit to society and not the individual whose intellectual property might be protected. Furthermore, let us suppose also that there are clear personal benefits to open source software. Nevertheless, it has not been widely agreed that the facts concerning software Stallman cites do not establish his conclusion that, on balance, enforcement of copyright for software harms society. After all, it is not clear that, for example, Apple is on balance harming society by releasing only compiled code for its operating system to the public.

To be clear, it is true that open source software can be easily improved and exchanged by the public at large than closed source software can. However, there are possible incentives *for society* in having closed source software. As we have been stressing, some software projects are extremely large and expensive. By enforcing copyright for software, the public can ensure that companies that take great risk in creating a piece of software can have a reasonable expectation of recouping their expenses. Perhaps the benefit of having Apple invest considerable resources into its operating system, OS X, with the expectation of profit outweighs the harms of my not being able to (legally) copy and alter it. It is clear, then, that the questions of if and how much of software ought to be open source remains itself open. In the next section we show that new arguments concerning formal methods can be applied to this issue, and in a way that may tip the balance for certain uses of software.

5.2 Formal Methods and Free Software

Consider the following claim: the specification and verification of software depends on publicly scrutable proofs and to be accomplished effectively, it must be done in an open intellectual community. De Millo, Lipton, and Perlis have famously argued that mathematical proof is a *social process*, meaning that the value of a proof is to convince other mathematicians of the truth of the proved fact, and indeed, to say that a fact is proved is to say that the community has internalized the truth of the fact, not that some informal logical derivation has been published (however, they claim this does *not* hold for formal verification specifically; we address that claim shortly) (Millo et al. 1979).

Couched in terms of Stallman's freedoms (Section 5.1), the motivation for open proofs mostly falls under Freedom 1 (freedom to study how the proof works) and Freedom 3 (freedom to improve the proof). Presently, open, independently verifiable results are not ubiquitous in formal methods specifically or computer science more generally. Indeed, Denning argues that computer science is not perceived as—and in some ways, does not act as—a science (Denning 2005).

For example, Denning cites a study showing that 50% of the published models and hypotheses in computer science go untested (Tichy 1998).

A number of reasons exist for the current state of affairs. Some reasons include (1) computer science being a young field, (2) much research occurs in industrial labs of for-profit companies, which sometimes do not release internal intellectual property validating the published research, and (3) there is a lack of data on which to validate models and approaches (in formal verification, data are programs and designs, many of which are closed-source). The full set of causes are surely complex and are a combination of economic, cultural, and political reasons.

Regardless of the reasons for why results are not publicized more broadly, the reasons for why they *should* be publicized are not different than the "four-freedoms justification" for why software should be open. For a concrete anecdotal reason, consider the following: A theory of a class of distributed protocols was developed, verified, and subsequently published (Rushby 1999). One of us read the published paper but was slightly unclear regarding a few of the details—the theory was only informally presented in the published paper. Fortunately, the original specifications were made publicly available, and we able to examine them. In doing so, we found that three out of four of the fundamental system assumptions (i.e., axioms in the formal theory) not only failed to model the domain of discourse but were in fact inconsistent. We were able to quickly correct the theory—the fundamental insights were correct—and publish the corrections (Pike 2006).

If the original specifications and proofs had not been available, the inconsistencies would have remained. Nevertheless, merely making them public was also not enough: in this instance, the original paper had been cited two or three dozen times, and it had been reviewed multiple times. None of those citing the paper or reviewing it caught the inconsistencies (indeed, we had cited the paper in earlier work without inspecting the proofs themselves). With open software, there is the presupposition or hope that when it gets used, bugs will be caught and corrected. Perhaps it is the same with formal proofs (we believe we were the first to "use" the published proofs in this particular instance).

De Millo, Lipton, and Perlis contradict this claim (Millo et al. 1979). Essentially, they argue that unlike pure mathematics, formal proofs about software, even if they are free, will never garner the interest required to bring the value of independent review like for pure mathematics.

> Verifications are long and involved but shallow; that's what's wrong with them. ...Nobody is going to run into a friends' office with a program verification. Nobody is going to sketch a verification on a paper napkin. Nobody is going to buttonhole a colleague into listening to a verification. Nobody is going to read it.

Our anecdotal story above contradicts this claim. More generally, one of the present authors works in industry at a company that does formal verifications, and he is routinely buttonholed into listening to verifications. To be sure, he is not buttonholed into listening to the details of an entire verification but often key parts of a proof or new insights and techniques.

More generally, De Millo, Lipton, and Perlis make a few errors in their conclusions about the utility of open verifications. First, free and open verifications are important even if nobody manually reads them. Automated proof checkers exist today (De Millo et al.'s paper was written in 1979) so that one can check the correctness of a verification without reviewing it manually, but this still requires that the proofs be available. Second, more abstract concepts in computer science (e.g., the distributed protocol verification mentioned above) *are* short and simple enough to be manually reviewed. Third, "chicken-egg" phenomena may exist: nobody reviews free and open verifications, because there are no free and open verifications! Nearly 30 years ago, would anyone have predicted the success of free software? Would we have predicted the thousands of software develops who contribute to open-source software projects?

Finally, just like in free software, open verifications would allow society to share the considerable burden of formal methods. Linux is a full-featured free operating system built mostly by volunteers (indeed, you can buy computers today running Linux from general-market dealers like Wal-Mart). Linux competes favorably with proprietary extremely well-funded operating systems, like Windows. Linux's success is possible due to massively-distributed efforts. Similar efforts in verification can have an analogous effect. Indeed, this is already happen ing: when a researcher develops a new formal verification tool, one convincing way to demonstrate its effectiveness is to find bugs in free software (like Linux) using it (see, for example, work by Dawson Engler et al. (2000)).

Let us reemphasize our main point of this section: the topic of free software is an example of a central issue in computer ethics, and the ethical issues of formal methods both inform and are informed by the free software debate. While we have presented specific arguments for why formal verifications should be free, we only wish to convince the reader that the debate over their freedom is important and belongs in computer ethics, too.

6 Conclusion

Why have formal methods *not* entered into the discourse in computer ethics in a central way already? Is it because of the contrast between computers and other things that can harm human beings? For example, there are widely accepted moral obligations that go along with automobile manufacturing. They have to be recalled if a safety-critical defect is found. Cigarette manufacturers have had multi-million-dollar liability suits levied against them. They are held accountable for a product they knew to be dangerous. To be sure, the causes are not completely ethical; for example, an automobile manufacture has strong economic motivations for building safe vehicles. Still, the difference between the computer industry and other industries appears to be one of kind, not degree.

Why has the public perception of computers been different? Why may a software manufacturer attach a "non-warranty" to software, to which you, the reader, have likely assented upon initially booting a new computer you have purchase? Why have there been no multi-million dollar liability suits against software manufacturers? Are current practices in software engineering the best practices? Does there exist a moral imperative to increase the assurance of software via formal methods, even if there may be no immediate economical gain in doing so?

Recall the developments in bioethics from the late 1960s to the present. The health care system, as a result of an interdisciplinary critique by philosophers, lawyers, and computer scientists, moved from widespread paternalism to a recognition of the autonomy of the patient. A dialogue outside of the computing profession can contribute to raising of public awareness, which itself is necessary for computing professionals to satisfy the imperative to prove safety, where and when it is a justifiable imperative.

We don't take the foregoing considerations to provide an airtight case for the claim that all software ought to be open, or that there is indeed an obligation to use formal methods. Rather, we hope to have shown that considerations of formal methods properly belong to computer ethics. To summarize, formal methods belongs computer ethics for three reasons: first, on the account given of what the purpose and nature of computer ethics are, formal methods belongs there. Second, canonical content of computer ethics is connected to issues related to formal methods. Finally, normative questions ought to be answered within a dialogue that includes the wider community of computer *users*, not just computer producers, and there is evidence that practitioners of normative theory bear some responsibility for ensuring more global deliberation on issues of professional responsibility.

In closing, let us summarize our goals in this paper, which included arguing for the following:

1. Computers, considered as *automated formal systems*, can be made safe in a way which other artifacts cannot.
2. There are open philosophical problems in the applied ethics of formal methods. The computer ethics community must help address these problems.

In furthering these goals, we

1. described formal methods in enough detail for computer ethicists to begin ethics projects about the subject, and
2. showed that formal methods belongs to computer ethics for both general reasons and consideration of important canonical issues within computer ethics.

We opened this article with two quotes from two of the world's most influential computer scientists. The quote from Knuth is slightly tongue-in-cheek, but the point to be drawn from it is that proof can never replace testing a program on real data in the intended environment. Dijksta's quote is meant quite seriously to explain that correctness cannot be proved via testing. Taken together, the quotes delightfully expose the work required to provide an ethics of formal methods.

Endnotes

1. We are grateful to an anonymous referee for encouraging us to clarify this point.
2. The result of this deliberation directly affected the nature of medicine in the West. See, for example, Emanuel and Emanuel, 1992.
3. The phrase "formal methods" is used both to refer to a field of practice (in which case the noun phrase is used in the singular) and as a description of the various methods used in the field (in which case the noun phrase is used in the plural).
4. The authors thank Jeffrey M. Maddalon of the NASA Langley Research Center for pointing out that the analogy breaks down in a critical way. We describe the shortcoming in Section 4.1.2.
5. We wish to emphasize the difference between formalization and formulation, particularly because the two notions are sometimes conflated in the field. *Formalization* is the act of expressing a concept mathematically; for example, Peano formalized arithmetic and Euclid formalized planar geometry. *Formulation* is the act of expressing mathematics in a formal language; for example, proofs can be formulated in the sequent calculus. (Steve Johnson of Indiana University made this distinction clear to the second author.)
6. Sometimes, software can be modified to mask faults in hardware.
7. For example, the *Software Reliability Research* group at Microsoft Research (http://research.microsoft.com/srr/) builds tools and invents new techniques to assist Microsoft developers to build more robust systems.
8. A *microkernel* is a lightweight (and usually highly robust) operating system.
9. "GNU" is a "recursive" acronym that stands for "GNU's not UNIX," where UNIX is a famous operating system standard.
10. Available at http://www.gnu.org/philosophy/free-sw.html.

References

Baleani, M., Ferrari, A., Mangeruca, L., Sangiovanni- Vincentelli, A., Peri, M., and Pezzini, S. "Fault-tolerant Platforms for Automotive Safety-critical Applications." In *CASES '03: Proceedings of the 2003 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, 2003, 170-77.

Barwise, J. "Mathematical Proofs of Computer System Correctness." *Notices of the AMS* 36 (1989): 844-51.

Berry, D. M. and Wing, J. M. "Specification and Prototyping: Some Thoughts on Why They Are Successful." In *Proceedings of the International Joint Conference on Theory and Practice of Software Development (TAPSOFT), Vol. 2.* 1985, 117-128. LNCS.

Charette, R. N. "Why Software Fails." *IEEE Spectrum Online*. 2005. http://www.spectrum.ieee.org/sep05/1685.

de Moura, L., Owre, S., Ruess, H., Rushby, J., and Shankar, N. "The ICS Decision Procedures for Embedded Deduction." In *2nd International Joint Conference on Automated Reasoning (IJCAR)*, vol. 3097 of *Lecture Notes in Computer Science*. Cork, Ireland: Springer-Verlag, 2004, 218-22.

Denning, P. J. "Is Computer Science Science?" *Communications of the ACM*, 48, no. 4 (2005): 27-31. http://cs.gmu.edu/cne/pjd/PUBS/CACMcols/cacmApr05.pdf.

Dowie, M. "Pinto Madness." *Mother Jones Magazine* (1977). http://www.motherjones.com/news/feature/1977/09/ dowie.html.

Elphinstone, K., Klein, G., Derrin, P., Roscoe, T., and Heiser, G. "Towards a Practical, Verified Kernel." In *Proceedings of the 11th Workshop on Hot Topics in Operating Systems (HOTOS)*. 2007.

Emanuel, E. J. and Emanuel, L. L. "Four Models of the Physician-patient Relationship." *Journal of the American Model Association* 267, no. 16 (1992): 2221-26.

Engler, D., Chelf, B., Chou, A., and Hallem, S. "Checking System Rules Using System-specific, Programmer-written Compiler Extensions." In *OSDI'00: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*. USENIX Association. 2000.

Ermann, M. D. and Shauf, M. S., eds. *Computer, Ethics, and Society*, 3rd ed. Oxford University Press, 2003.

Fodor, J. A. "The Big Idea: Can There Be a Science of Mind?" *Times Literary Supplement*. 1990.

Halfhill, T. R. "The Truth behind the Pentium Bug." *BYTE Magazine* (1995).

Haugeland, J. *Artificial Intelligence: The Very Idea*. Bradford Books, 1989.

Haugeland, J. *Mind Design II*, chapter What is Mind Design? MIT Press, 1997.

James, H. F. "Program Verification: The Very Idea." *Communications of the ACM*, 31, no. 9 (1988): 1048-63.

Jetley, R., Iyer, S. P., and Jones, P. L. "A Formal Methods Approach to Medical Device Review." *Computer* 39, no. 4 (2006): 61-67.

Johnson, D. G. *Computer Ethics*, 3rd ed. Prentice Hall, 2001.

J.R. Burch, E.M. Clarke, K.L. McMillan, D.L. Dill, and L.J. Hwang. "Symbolic Model Checking: 1020 States and Beyond." In *Proceedings of the Fifth Annual IEEE Symposium on Logic in Computer Science.* Washington, D.C. IEEE Computer Society Press, 1990, 1-33.

Knight, J. C. "Software Challenges in Aviation Systems. 2002a. http://groups.inf.ed.ac.uk/safecomp/Download/safecomp2002/knight_safecomp2002.pdf.

Knight, J. C. "Software Challenges in Aviation Systems." In *SAFECOMP*, Lecture Notes in Computer Science. Springer, 2002b, 106-12.

Lahiri, S. K. and Seshia, S. A. "The UCLID Decision Procedure." In *CAV'04: Proceedings of the 16th International Conference on Computer Aided Verification*. 2004, 475-78.

utz, R. R. "Analyzing Software Requirements Errors in Safety-critical Embedded Systems." In *IEEE International Symposium on Requirements Engineering*. 1993, 126-33.

Millo, R. A. D., Lipton, R. J., and Perlis, A. J. "Social Processes and Proofs of Theorems and Programs." *Communications of the ACM* 22, no. 5 (1979): 271-80.

Neumann, P. G. "Using Formal Methods to Reduce Risks." *Communications of the ACM* 39, no. 7 (1996): 114.

Pike, L. "A Note on Inconsistent Axioms in Rushby's 'Systematic Formal Verification for Fault-tolerant Time-triggered Algorithms'." *IEEE Transactions on Software Engineering* 32, no. 5 (2006): 347-48. http://www.cs.indiana.edu/~lepike/pub_pages/time_triggered.html.

Pike, L., Shields, M., and Matthews, J. (2006). "A Verifying Core for a Cryptographic Language Compiler." In *Proceedings of the Sixth International Workshop on the ACL2 Theorem Prover and its Applications.* ACM Press, 2006, 1-10. http://www.cs.indiana.edu/~lepike/pub_pages/acl2.html.

Rapaport, W. "Can Programs Be Verified?" 2007. http://www.cse.buffalo.edu/~rapaport/510/canprogsbeverified.html.

Rushby, J. "Formal Methods and Digital Systems Validation for Airborne Systems." Technical Report CR–4551, NASA, 1993.

Rushby, J. "Systematic Formal Verification for Fault-tolerant Time-triggered Algorithms." *IEEE Transactions on Software Engineering*, 25, no. 5 (1999): 651-60.

Smith, B. C. "The Limits of Correctness." *SIGCAS Comput. Soc.* 14,15, no. 1-4 (1985): 18-26.

Stallman, R. M. *The GNU Manifesto*. Cambridge University Press, 2003. Originally published by the Free Software Foundation, 1985.

Tavani, H. T. *Ethics and Technology: Ethical Issues in an Age of Information and Communication Technology*, 2nd ed. Wiley, 2007.

Tichy, W. F. "Should Computer Scientists Experiment More?" *Computer* 31, no. 5 (1998): 32-40.

Wiedijk, F., ed. *The Seventeen Provers of the World*, Foreword by Dana S. Scott, volume 3600 of *Lecture Notes in Computer Science*. Springer, 2006.

Winston, M. E. and Edelbach, R. D., eds. *Society, Ethics, and Technology*, 3rd ed. Thomson Wadsworth, 2006.

*An Algorithmic Approach to Information and Meaning: A Formal Framework for a Philosophical Discussion**

Hector Zenil
Institut d'Histoire et de Philosophie des Sciences et des Techniques (Paris 1/ENS Ulm/CNRS)

Abstract

I will survey some matters of relevance to a philosophical discussion of information, taking into account developments in algorithmic information theory (AIT). I will propose that meaning is deep in the sense of Bennett's logical depth, and that algorithmic probability may provide the stability needed for a robust algorithmic definition of meaning, one that takes into consideration the interpretation and the recipient's own knowledge encoded in the story attached to a message.

Keywords: information content; meaning; algorithmic probability; algorithmic complexity; logical depth; philosophy of information; information theory.

Introduction

Information can be a cornerstone for interpreting all manner of phenomena, as it can constitute the basis for a description of objects. While it is legitimate to study ideas and concepts related to information in their broadest sense, that the use of information outside formal contexts amounts to misuse cannot and should not be overlooked. It is not unusual to come across surveys and volumes devoted to information (in the larger sense) in which the mathematical discussion does not venture beyond the state of the field as Shannon (1948) left it some 60 years ago. Recent breakthroughs in the development of information theory in its algorithmic form—both theoretical and empirical developments possessing applications in diverse domains (e.g., Li and Vitányi 2008; Li et al. 2001; Li et al. 2003; Zenil et al. forthcoming)—are often overlooked in the semantic study of information, and it is philosophy and logic (e.g., epistemic temporal logic) that are resorted to in attempting to account for what is said to be the semantic formalism of information. As examples one may cite the work of Floridi (2003 and 2008) and Godfrey-Smith and Sterelny (2008). In the best of cases, algorithmic information theory (AIT) is not given due weight. Its basic definitions are sometimes inaccurately rendered (e.g., the incomplete definition of Bennett's logical depth [Bennett 1988] in Roederer 2010, p. 25).

In Floridi 2008, for example, the only reference to AIT as a formal context for the discussion of information content and meaning is a negative one—appearing in van Benthem's contribution (Floridi 2008, p. 171). It reads:

> To me, the idea that one can measure information flow one-dimensionally in terms of a number of bits, or some other measure, seems patently absurd...

I think this position is misguided. When Descartes transformed the notion of space into an infinite set of ordered numbers (coordinates), he did not deprive the discussion and study of space of any of its interest. On the contrary, he advanced and expanded the philosophical discussion to encompass concepts such as dimension and curvature-- which would not have been possible without the Cartesian intervention. Perhaps this answers the question that Benthem poses immediately after the above remark (Floridi 2008, p. 171):

> But in reality, this quantitative approach is spectacularly more successful, often much more so than anything produced in my world of logic and semantics. Why?

Accepting a formal framework such as algorithmic complexity for information content does not mean that the philosophical discussion of information will be reduced to a discussion of the numbers involved, as it did not in the case of the philosophy of geometry after Descartes.

The foundational thesis upon which information theory rests today (derived from Shannon's work) is that information can be reduced to a sequence of symbols. Despite the possibility of legitimate discussions of information on the basis of different foundational hypotheses, in its purely syntactic variant, information theory can be considered in large part achieved by Shannon's theory of communication (see Box 1).

Epistemological discussions are, however, impossible to conceive of in the absence of a notion of semantics. Much work has been done in logic to capture the concept of meaning in a broader and formal sense. However, little or nothing has been done to explain meaning using pure computational models—whether to extend Shannon's work on information, to explain meaning in light of Turing's merging of information and computation, or to explain meaning in light of current developments, as epitomized by the theory of AIT.

Box 1. Shannon's entropy is defined as a measure of the average information content associated with a random outcome. Formally, Let $d=(p_1, p_2, ..., p_n)$ be a finite discrete probability distribution. That is, suppose $p_k \geq 0$ for $k=1, 2, ..., n$ and $\sum^n_{k=1} p_k = 1$. The uncertainty concerning a possible outcome with probabilities $p_1, p_2, ..., p_n$ is called the entropy of the distribution P and is measured by $H(P)=H(p_1, p_2, ..., p_n)$ as introduced by Shannon (1948) and defined by $H(p_1, p_2, ..., p_n) = \sum^n_{k=1} p_k \log_2 1/p_k$. It indicates how many bits are required to encode a message in order to send it through a communication channel with minimum capacity. It counts how many different symbols a message has, weighted by their probability distribution. It is therefore a measure of diversity, not of order or disorder.

Semantics is concerned with content. Both the syntactic and semantic components of information theory are concerned with order, the former particularly with the number of symbols and their combinations, while the latter is intimately related to structure. The theory of computation is the context provided by the theory of algorithmic information for discussion of the concept of information. Within this context the description of a message is interpreted in terms of a program. The following sections are an overview of the different formal directions in which information has developed in the last decades. They leave plenty of room for fruitful philosophical discussion, discussion focusing on information per se as well as on its connections to aspects of physical reality.

2 Communication, diversity and complexity

Shannon's conception of information inherits the pitfalls of probability (see Box 1). Which is to say that one cannot talk about the information content of individual strings. However, misinterpretations have dogged Shannon's information measure from the inception, especially around the use of the term *entropy*, as Shannon himself acknowledged. The problem has been that Shannon's entropy is taken to be a measure of order (or disorder), as if it were a complexity measure (analogous to physical entropy in classical thermodynamics). Shannon acknowledges that his theory is a theory of communication and transmission and not one of information (Shannon 1948).

Unlike algorithmic complexity (see Box 2) Shannon's entropy does not take into consideration the internal structure of the message. Consider the strings 0101010101 and 0100101100. They both have exactly the same Shannon entropy because the number of occurrences of 1 and of 0 is the same in both strings. As a second example, consider the sentence "A quick brown fox jumps over the lazy dog" and the scrambled version "rl y feawhojkouq A vdpegsxioz r cmunotb." They both have the same Shannon entropy value even though clearly one tells us something while the other is nonsensical. Shannon's Entropy basically says that because the members of each of these pairs of messages use about the same number of symbols, one needs a communication channel of about the same size for both members of the pair.

That Shannon's measure is computable and easily calculable in practice may account for its frequent and unreasonable application as a complexity measure. The fact that algorithmic complexity is not computable, however, doesn't mean that one cannot approximate it.

Shannon's approach doesn't help to define information content or meaning. For example, think of a number like π which is believed to be normal (that is, that its digits are equally distributed), and therefore has little or no redundancy. The number π has no repeating pattern (because it is an irrational number). Lacking a pattern, there is no way to optimize a channel through which to transmit it. π, however, can be greatly compressed using any of the known briefly describable formulas generating its digits, so that one can send the formula rather than the digits. But this kind of optimization is not within the scope of Shannon's communication theory. Unlike Shannon's treatment of π, one can think of π as a meaningful number because of what it represents: the relationship between any circumference and its diameter. I will argue that meaning can be treated formally, using concepts of AIT to approach these matters.

The main point of relevance to us made by Shannon when formulating his measure in the context of communication is that in practice a message with no redundancy is more likely to carry information if one assumes one is transmitting more than just random bits. If something is random-looking, then it will usually be considered meaningless. To say that something is meaningful usually implies that one can somehow arrive at a conclusion based on it. Information has meaning only if it has a context, a story behind it. Meaning, in a causal world, is the story attached to a message.

Box 2. Algorithmic complexity is the length in bits of the shortest program producing a string s when running a program p on a universal Turing machine U upon halting. One refers to $C(s)$ as the algorithmic complexity of s. Formally, $C(s)=|p| : U(p)=s$ where $|p|$ is the length of p measured in bits. Algorithmic complexity formalizes the concept of simplicity versus complexity. For an introduction to AIT, please see Calude and Stay 2006, and Li and Vitányi 2008.

3 Data + program is message + interpretation

Among the several contributions made by Alan Turing on the basis of his concept of computational universality is the unification of the concepts of data and program. Turing machines are extremely basic abstract symbol-manipulating devices, which despite their simplicity, can be adapted to simulate the logic of any computer that could possibly be constructed. While one can think of a Turing machine input as data, and a Turing machine rule table as its program, each of them being separate entities, they are in fact interchangeable as a consequence of universal computation, as shown by Turing himself, since for any input s for a Turing machine M, one can construct M' with empty input such that M and M' accept the same language, with M' a (universal) Turing machine accepting an encoding of M as input and emulating it for an input s for M in M'.

In other words, one can always embed data as part of the rule table of another machine. The identification of something as data or a program is therefore merely a customary convention and not a fundamental distinction. This is noteworthy because though it may seem that a message is to data as an interpretation is to a program, part of the argument is that the message cannot be formally captured in both computational and semantic terms if the interpretation is not part of it. So even if on the surface we may make a distinction, just as we make a distinction between a message and its recipient in the real world, the difference is not essential. Hence subsuming everything under the heading of "message" will allow us to define the concept of *meaning* on the basis of the theory of computation and algorithmic information. For example, messages (inputs) for which a Turing machine halts can be taken to be messages that have been understood. It is certain that in the case of a universal Turing machine there will be messages for which the machine will halt and others for which it will not halt, a desirable property insofar as we are concerned with defining meaning. Which is to say that no matter what the meaning of a message, there should always be recipients that are capable of interpreting it and others that are not. That some *understand* a message by halting (i.e., conventionally) doesn't mean, however, that all of them *understand* it in the same way, since each may react to it in a different way—which is desirable insofar as we are seeking to define a concept of subjective meaning. These simple first conventions will allow us to apply several concepts from algorithmic information theory, notably the concepts of conditional complexity, algorithmic probability and logical depth, in order to impart sense and robustness to our algorithmic approach to the concept of *meaning* from a computational perspective.

4 Information content and meaning

Nevertheless, Shannon's notion of information makes it clear that information content is subjective (Shannon himself):

> Frequently the messages have meaning: that is they are referred to or correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages. (Shannon 1948)

Subjective doesn't mean, however, that one cannot define information content formally, only that one should include a plausible interpretation in the definition, a point we will explore in the next section.

Shannon's contribution is seminal in that he defined the bit as the basic unit of information, as do our best current theories of informational complexity. Any sequence of symbols can be translated into a binary sequence, thereby preserving the original content, as it can be translated back and forth from the original to the binary and vice versa. Shannon's information theory approaches information syntactically: whether and how much information (rather than what information) is conveyed. And as a physical phenomenon: the basic idea is to make the communication channel more efficient.

Unlike Shannon's entropy, algorithmic complexity considers individual objects independent of any probability distribution. For example, different initial segments of the Fibonacci sequence have each a different entropy. Each of the 10 segments having 10 more values of the Fibonacci sequence correspond to the following sequence of Shannon's entropy values: 2.163, 2.926, 3.354, 3.654, 3.88, 4.071, 4.228, 4.36, 4.484, 4.591—simply because the longer the sequence of numbers, the more bits are required to encode them as such through a communication channel. Yet one can decide that in order to recover the same message at about the same size, it would be preferable to send the formula together with the number of Fibonacci numbers to generate. The sequence can be expressed as $F(n) = F(n-1) + F(n-2)$ for $n=\{3,4,5,...\}$, $F(1)=1$, $F(2)=1$. Which as a formula has a low and constant algorithmic complexity for any segment of the Fibonacci sequence. This example reveals that this is a significant difference: two messages of arbitrarily different algorithmic complexity can have the same Shannon entropy. Shannon's entropy represents an absolute limit on the best possible lossless compression of the algorithmic complexity, i.e., $C(s)$ (program-size algorithmic complexity) cannot be smaller than $H(s)$ (Shannon's entropy). However, the Fibonacci sequence as a message has the same meaning for algorithmic complexity, again another desirable feature of algorithmic complexity (and clearly one that would be unavailable were the problem to be approached through Shannon's entropy). Further information about Shannon's entropy in comparison to algorithmic complexity is available in Grünwald and Vitányi 2004.

5 Information content and intrinsic meaning

As an attempt to define lack of meaning, think of a single bit. A single bit does not carry any information, and so it cannot but be meaningless if there is no recipient to interpret it as something richer. The Shannon entropy of a single bit is 0 because one cannot establish a communication channel—no message can be encoded or transmitted with a single letter of the alphabet. It is intrinsically meaningless because there is no context. The same is true for a string of *n* identical bits (either 1s or 0s). To give it a meaning one would likely be forced to make an external interpretation, because even if it carries a message it cannot be intrinsically very rich, simply because it cannot carry much information. In both cases Shannon's entropy and the algorithmic complexity of such a string is very low.

At the other extreme, a random string cannot be usually considered meaningful. What one can say with certainty is that something meaningful should therefore lie between these two extremes: *no information* (trivial) or *complete nonsense* (random). Algorithmic complexity (see Box 2) opposes what is simple to what is complex or random and can be thought of as a first approximation to meaning, as opposing what is meaningful in some rough sense to what is meaningless. However, algorithmic complexity alone does not suffice to define meaning because it cannot properly oppose complexity to randomness, only simplicity to complexity or randomness. This is because algorithmic complexity associates randomness with the highest level of complexity. But using Bennett's logical depth (1988) (also rooted on algorithmic complexity) we are able to distinguish between something that looks organized and something that looks random or trivial by introducing time (a parameter that seems unavoidable in the unfolding of a message, which makes it reasonable to associate this measure with a measure of *physical complexity* as Bennett himself suggests [Bennett 1990]). This connection to Bennett's logical depth means that a message cannot be instantaneously meaningful if it is not in a proper context.

It is generally accepted that meaning is imparted by the observer, that it is the interpretation of the message by its recipient. In order for the information conveyed to have any semantic value, it must in some manner add to the knowledge of the receiver. I claim that when it comes to mapping meaning onto information content, these semantic properties can be accounted for using concepts such as conditional algorithmic complexity (see Box 3) and logical depth (see Box 4).

For example, conditional algorithmic complexity (see Box 3) can define a distance measure between messages. It is clear from the conditional definition that even if a string s is meaningless, the complexity of s given s is very low because the length of the shortest program producing s with input $s$is the shortest possible.

Box 3. Conditional algorithmic complexity is defined (Li and Vitányi 2008) as the shortest program p, for which the universal Turing machine U outputs s given x, that is $C(s|x) = \min|p| : U(x,p) = s$. Notice that the definition of algorithmic complexity in Box 2 is a special case of the conditional one when there is no x, i.e., $C(s)=C(s|\varepsilon)$, that is the algorithmic complexity of s given no other information.

6 Meaning is logically deep

As is well known, the problem with meaning is that it is highly dependent on the recipient and its interpretation. Connecting meaning to the concept of logical depth has the advantage of taking into account the story and context of a message, and therefore of potentially accounting for the likely recipient's interpretation. A meaningful message (short or long) contains a long computational history when taken together with the associated computation, otherwise it has little or no meaning. Hence the pertinence of the introduction of logical depth.

Think of a winning number in a lottery. The number by itself may be meaningless for a recipient, but if two parties had shared information on how to interpret it, the information shared beforehand becomes part of the computational history and as such not unrelated to the subsequent message. The only way to interpret a number as being the winning number of a lottery is to have a story, not just a story that relates the number to a process, but one that narrates the process itself. Since winning a prize is no longer a matter of apparent chance but has to do with the release of information (both the number and the interpretation of the number), it is therefore not the number alone that represents the content and meaning of the message (the number), but the story attached to it.

Box 4. Bennett's logical depth is defined (Bennett1988) as the execution time required to generate a string by a near-incompressible program, i.e., one not produced by a significantly shorter program. Logically deep objects contain internal evidence of having been the result of a long computation and satisfy a slow-growth law (by definition).

There are also messages that contain the story in themselves. If instead of a given number one substitutes the interpretation of such a number, the message can be considered meaningful in isolation. But both cases have the same logical depth, as they have the same output and computing time and are the result of the same history (even if in the first case such a history may be rendered in two separate steps) and origin, hence the definition seems robust enough.

If a Turing machine randomly performs a lot of work when provided with a random input, making it look *meaningful*, it may seem that our approach is not robust enough, since something that's taken to be meaningful is actually just a random computation. There are, however, two possible answers to this objection: On the one hand, the probability of a machine undertaking a long computation by chance is very low. Calude and Stay (2006) prove that of machines that halt, most will halt after a few steps. This happens for most strings, meaning that most messages are meaningless if both the message and the computation do not somehow *resonate* with each other, which recalls an intuitive requirement for considering something a meaningful message. On the other hand, also based on the results of Calude and Stay, algorithmic probability almost guarantees the non-existence of *Rube Goldberg* machines (amusing machines that do a lot of work in order to accomplish a trivial task), which implies that the amount of work performed (the number of steps and output length) by a Turing machine that halts will be proportional to the information content of the input message.

7 An algorithmically robust definition of meaning

Algorithmic probability, as defined by Solomonoff (1964) and Levin (1974) (see Box 5) indicates that every outcome is likely to be produced by the shortest program(s) producing that outcome. In other words, meaningful messages would have little chance of being interpreted as such by chance.

It is algorithmic probability that would account for the robustness of this algorithmic approach to meaning. The

Box 5. Algorithmic probability: The algorithmic probability of a string s is the probability of producing s with a random program p (a sequence of fair coin flip inputs) when running on a universal prefix-free Turing machine (Levin 1974). That is, a machine for which a valid program is never the beginning of any other program, so that one can define a convergent probability the sum of which is at most 1. Formally, $m(s) = \sum_{p:U(p)=s} 2^{-|p|}$, i.e., the sum over all the programs for which U with p outputs the string s and halts.

meaning of a message only makes sense for particular recipients (not for any random ones). A message that has meaning for someone may not have meaning for someone else—just the kind of property one would desire in a concept entailing the meaning of *meaning*. This is what happens when some machines react to a *meaningful* input rather than to a *random* one. Algorithmic probability guarantees that most machines will halt almost immediately with no computational history for a given message. In other words, there is a correspondence between a meaningful input, computation time and a structured outcome, given the connection between algorithmic probability and algorithmic complexity (see Box 6).

Box 6. The coding theorem is a theorem connecting algorithmic probability to algorithmic complexity. Algorithmic probability is related to algorithmic complexity in that $m(s)$ is at least the maximum term in the summation of programs given that it is the shortest program that has the greater weight in the summation of the fractions defining $m(s)$. Formally, the theorem states that the following relation holds: $-\log_2 m(s) = C(s) + O(1)$. For technical details see Calude and Stay 2006.

8 Finite randomness (just like meaning) is in the eye of the beholder

In a move that parallels the mistaken use and overuse of Shannon's measure as a measure of complexity, the notion of complexity is frequently associated, in the field of complex systems, with the number of interacting elements or the number of layers of a system. Researchers who make such an association should continue using Shannon's entropy since it quantifies the distribution of elements, but they should also be aware that they are not measuring the complexity of a system or object, but rather its diversity, which may be a different thing altogether (despite being roughly related).

As has been shown by Stephen Wolfram, it is not always the case that the greater the number of elements the greater the complexity, nor is it the case that a greater number of layers or interactions make for greater complexity, for the simplest computing systems are capable of the greatest apparent complexity (Wolfram 2002).

The theory of algorithmic randomness does not guarantee that a string of finite length cannot be algorithmically compressed. Nonetheless, any string is guaranteed to occur as a substring (with equal probability) in any algorithmically random infinite sequence. But this has to do with the semantic value of AIT, given that a finite string has meaning only in a particular context, as a substring of a larger, potentially longer and essentially different string. Therefore, one can declare a string to be random-looking only as long as it does not appear as a substring embedded in another finite or infinite string.

A string may be *random* at the scale of an infinite sequence, but since all possible strings are contained in an infinite random sequence, random blocks of strings may not always look random. For example, consider the following pseudo-randomly generated string (generated with *Mathematica's* default random number generator: 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1. This string looks quite random when taken as a whole, but it actually contains several blocks that do not look random at all (e.g., 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 from position 19 to position 29; or 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 from position 30 to position 46) if isolated.

On the other hand random-looking strings can always be part of larger non-random sequences. A string with period 10 (and therefore compressible to about a tenth of its size): 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0; contains, if isolated, a "random-looking" string 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0.

The same goes for a message. A message can be contained in a longer message, and actually its interpretation may change from one context to another. From the perspective of algorithmic complexity (when interpreted in terms of meaning) this property can be seen as an indication that there cannot be an ultimate interpretation of the meaning of a message—yet another desirable feature of algorithmic complexity that mirrors our intuitive sense of the concept of *meaning*.

Box 7. The Halting problem and Chaitin's Ω: As widely known, the Halting problem for Turing machines is the problem of deciding whether an arbitrary Turing machine T eventually halts on an arbitrary input s. Halting computations can be recognized by simply running them for the time they take to halt. The problem is to detect non-halting programs, about which one cannot know if the program will run forever or will eventually halt. An elegant and concise representation of the halting problem is Chaitin's irrational number Ω (Chaitin 1975), defined as the halting probability of a universal computer programmed by coin tossing. Formally, $0 < \Omega = \sum_{p\,halts} 2^{-|p|} < 1$ with $|p|$ the size of p in bits. Chaitin's Ω number is the halting probability of a universal (prefix-free) Turing machine running a random program (a sequence of fair coin flip bits taken as a program).

At the other extreme, in this algorithmic context, there is Chaitin's Ω number (Chaitin 1975) that may be regarded as entailing the greatest possible meaning because it encodes all possible messages in the form of answers to all possible questions encoded by Turing machines. That Chaitin's Ω is in practice inaccessible seems desirable if we are to avoid a contradiction in the concept of meaning, and also in light of the fact that one cannot expect to encode all meanings in a single message. In other words, complete meaningfulness or the apprehension of all possible meanings is unattainable via this algorithmic approach, as it is ultimately uncomputable. Which is just what one would intuitively expect to be the case with *meaning* in the broadest sense.

9 Towards a philosophical agenda

The previous discussion sketches a possible agenda for a philosophy of information in the context of the current state of the theory of algorithmic information. Focusing more on the core of the theory itself, there are several directions that a deeper exploration of the foundations of algorithmic information might take.

There is, for example, Levin's contribution (Levin 1974) to algorithmic complexity in the form of the eponymous semi-measure, motivated by a desire to fundamentally amend Kolmogorov's plain definition of complexity in light of the realization that information should follow a law of non-growth conservation. This is an apparently different motivation from the one behind Gregory Chaitin's definition of algorithmic complexity in its prefix-free version.

There are also laws of symmetry and mutual information discovered by Gács (1974), and Li and Vitányi (2008), for example, which remain to be explored, fully understood and philosophically dissected.

Furthermore there are the subtle but important differences involved in capturing organization in information through the use of algorithmic randomness versus doing so using Bennett's logical depth (Bennett 1988), a matter brought to our attention in, for example, Delahaye 2009. The motivation behind Bennett's formulation of his concept of logical depth was to capture the notion of complexity by taking into account the history of an object. It had, in our algorithmic approach to information, the important consequence of classifying intuitively shallow physical objects as objects deprived of meaning.

There is also the question of the dependence of the definitions on the context in which meaning is evaluated (the choice of universal Turing machine), up to an additive constant, a question which has recently been addressed by Delahaye and Zenil (2007 and 2011), who propose that reasonable choices of computational formalisms actually lead to reasonable evaluations of complexity (Delahaye and Zenil 2008). This is the problem of finding a stable framework for a robust enough evaluation of information content.

In Floridi 2008, van Benthem highlights an issue of great philosophical interest when he expresses a desire to understand the *unreasonable effectiveness* (a phrase he claims is borrowed) of quantitative information theories.

Paradoxically, my concern would be with the *unreasonable ineffectiveness* of qualitative information theories, notably algorithmic complexity, given that it is the *unreasonable effectiveness* of quantitative information theories, notably Shannon's notion of information entropy, that has mistakenly led researchers to use it in frameworks in which a true (and universal) measure of complexity is needed. The connections between Shannon's entropy and Kolmogorov complexity are investigated in detail in Grünwald and Vitányi 2004.

It is the ineffectiveness of algorithmic complexity that imbues information content with its deepest character, given that its full characterization cannot effectively be achieved even if it can be precisely defined. Hence Van Benthem opens up a rich vein, this discussion being potentially fruitful and of great interest even if it has hitherto been largely ignored. I think the provisional formulations of the laws of information, together with their underlying motivations should be a central part of the discussion, if not the main focus of a semantic approach to information which is more integrally rooted in current mathematical developments.

It may be objected that a semantic approach to the study of information should not be reduced to the algorithmic, which is sometimes considered a syntactic-digital view of information. That, however, would be rather an odd objection, given that most texts on information start with Shannon's information theory, without however taking the next natural step and undertaking a discussion of the current state of information as exemplified by AIT. So either the philosophy of information ought to take a completely different path from Shannon's, which inevitably leads to the current state of algorithmic information and prompts deeper exploration of it, or else it should steer clear of the algorithmic side as being separate from and alien to it.[1] In other words, I don't find it consistent to cover Shannon's work while leaving out all further developments of the field by, among others, Kolmogorov, Chaitin, Solomonoff, Levin, Bennett, Gács, and Landauer. As I have pointed out in the previous section, there is a legitimate agenda concerning what some may call the syntactic-mechanistic branch of the study of Information, which, paradoxically, I think is the most interesting and fruitful part of the semantic investigation, and one that mainstream Philosophy of Information has traditionally steered clear of for the most part, a few exceptions notwithstanding (Parrochia 1996; McAllister 2003).

No account of what information might be can be considered complete without taking into account the interpretations of quantum information. One relevant issue has been raised by Wheeler (1990), though perhaps with reference to a different scale, and that is whether an observer is necessary for information to exist and for an observation to have meaning. There does not exist a universally accepted interpretation of quantum mechanics, although the so-called *Copenhagen Interpretation* is considered the mainstream one. Discussions about the meaning of quantum mechanics and its implications do not, however, lead to a consensus. It is beyond the scope of this paper to further discuss the quantum approach other than to point out its pertinence in an encompassing discussion. See, for instance, Cabello and Joosten 2011.

9.1 Basics to agree upon

We should agree upon fundamental properties of information derived from the current state of AIT that, as I've argued, can serve as a basis for a mathematical framework in a philosophical discussion. Although this is not the place to discuss the several results of the theory of algorithmic complexity, here is a non-exhaustive list of some of the points to be agreed upon, together with the claims that meaning can be captured and at least some of its properties studied. The list, in a certain order of logical derivation, is:

- The basic unit of information is the bit, but information remains subjective (Shannon 1948).
- Shannon's information measure cannot capture content, organization or meaning as it is neither a measure of information content nor of complexity.
- Algorithmic complexity is an objective and universal framework for capturing structure and randomness.
- Randomness implies the impossibility of information extraction (Chaitin 1975).
- Shallowness is meaningless (Kolmogorov 1965).
- Randomness is meaningless (Bennett 1988).
- There are strong connections between logical and thermodynamic (ir)reversibility to be explored (Bennett 1987, Fredkin 1982, Toffoli 1982).
- Information can be transformed into energy and energy into information (Landauer 1961, Bennett 1988).
- Information follows fundamental laws: symmetry, non-growth, mutual information and (ir)reversibility (Gács 1974, Zvonkin 1970, Levin 1974, Bennett 1987, Landauer 1961).
- Matter and information are deeply connected, although it is an open question which of the two is more fundamental (Wheeler 1990, Feynman 1996, Bennett 1987, Landauer 1961, Fredkin 1982, Wolfram 2002).

Information has also begun playing a major role in the interpretation of quantum mechanics (Lloyd 2006; Cabello and Joosten 2011) and is assuming foundational status in some models of modern physics (Svozil 1994) as it did in classical physics, notably in thermodynamics, and more recently in cosmology (Bekenstein 2003). We shall further survey the listed properties and the connections of information to physics in a follow-up to this paper.

10 Concluding remarks

A common language and a commonly agreed upon formal framework seem to be necessary. I've claimed that algorithmic information is suitable for defining individual information content and for providing a characterization of the concept of meaning in terms of logical depth and algorithmic probability. This rather formal computational characterization does not mean that a discussion of algorithmic information would be deprived of legitimate philosophical interest.

I have briefly drawn attention to and discussed some of the questions germane to a philosophy of algorithmic information in connection to a semantic definition of meaning. I've argued that algorithmic complexity and algorithmic probability taken together with Bennett's logical depth can constitute an appropriate computational framework within which to discuss information content and meaning. Nor does the fact that meaning can be fully formalized mean either that it will lose its most valuable characteristics, such as subjectivity with respect to a recipient. Such a subjective and rich dimension can be computationally grasped as proposed herein, and can constitute another point of departure for an organized philosophical discussion accounting for and covering a field that can no longer be ignored in philosophical discussions of information.

*Presented at the *Interdisciplinary Workshop: Ontological, Epistemological and Methodological Aspects of Computer Science*, Philosophy of Simulation (SimTech Cluster of Excellence), Institute of Philosophy, Faculty of Informatics, University of Stuttgart, Germany, July 7, 2011.

Acknowledgments

The author would like to thank Peter Boltuc for his suggestions, and Selmer Bringsjord for his comments, objections and valuable views on information from his more purely logical perspective. Selmer and I may not agree but we agree to disagree.

Endnote

1. Pieter Adriaans has presented similar arguments (Adriaans 2010) in relation to the often mistaken tendency of semantic approaches to information to largely ignore the theory of algorithmic information.

References

P. Adriaans. "A Critical Analysis of Floridi's Theory of Semantic Information." *Knowlegde, Technology and Policy* 23 (2010): 41-56.

J.D. Bekenstein. "Information in the Holographic Universe." *Scientific American* 289, no. 2 (2003): 61.

C.H. Bennett. "Logical Depth and Physical Complexity." In *The Universal Turing Machine--a Half-Century Survey*, edited by Rolf Herken. Oxford University Press, 1988, 227-57.

C.H. Bennett. "How to Define Complexity in Physics and Why." In *Complexity, Entropy and the Physics of Information*, edited by W.H. Zurek. Addison-Wesley, SFI studies in the sciences of complexity, 1990, 137-48.

C.H. Bennett. "Demons, Engines and the Second Law." *Scientific American* 257 (1987): 108-16.

A. Cabello, J.J. Joosten. "Hidden Variables Simulating Quantum Contextuality Increasingly Violate the Holevo Bound." In *Unconventional Computation 2011* (Lecture Notes in Computer Science 6714), edited by C.S. Calude et al. Springer 2011.

C.S. Calude and M.A. "Stay, Most Programs Stop Quickly or Never Halt." arXiv:cs/0610153v4 [cs.IT], 2006.

G.J. Chaitin. "A Theory of Program Size Formally Identical to Information Theory." *J. Assoc. Comput. Mach.* 22 (1975): 329-40.

J.P. Delahaye and H. Zenil. "On the Kolmogorov-Chaitin Complexity for Short Sequences." In *Complexity and Randomness: From Leibniz to Chaitin*, edited by Cristian Calude. World Scientific, 2007.

J.-P. Delahaye and H. Zenil. "Towards a Stable Definition of Kolmogorov-Chaitin Complexity." arXiv:0804.3459v3 [cs.IT], 2008.

J.-P. Delahaye. *Complexité aléatoire et complexité organisée.* Editions Quae, 2009.

J.-P. Delahaye and H. Zenil. "Numerical Evaluation of the Complexity of Short Strings: A Glance Into the Innermost Structure of Algorithmic Randomness." arXiv:1101.4795v2 [cs.IT], 2011.

R. P. Feynman, J.G. Hey, and W. Allen. *Feynman Lectures on Computation*. Addison-Wesley, 1996.

L. Floridi. *The Blackwell Guide to the Philosophy of Computing and Information (Blackwell Philosophy Guides)*. Wiley-Blackwell, 2003.

L. Floridi, ed. *Philosophy of Computing and Information: 5 Questions.* Automatic Press / VIP, 2008.

E. Fredkin and T. Toffoli. "Conservative Logic." *International Journal of Theoretical Physics* 21 (1982): 219-53.

P. Gács. "On the Symmetry of Algorithmic Information." *Soviet Mathematics Doklady* 15 (1974): 1477-80.

P. Grünwald and P. Vitányi. "Shannon Information and Kolmogorov Complexity." *Computing Research Repository - CORR*, 2004.

A. N. Kolmogorov. "Three Approaches to the Quantitative Definition of Information." *Problems of Information and Transmission* 1, n. 1 (1965): 1-7.

R. Landauer. "Irreversibility and Heat Generation in the Computing Process." *IBM Journal of Research and Development* 5 (1961): 183-91.

L. Levin. "Laws of Information Conservation (Non-growth) and Aspects of the Foundation of Probability Theory." *Problems of Information Transmission* 10, no. 3 (1974): 206-10.

M. Li, P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed. Springer, 2008.

M. Li, J. Badger, X. Chen, S. Kwong, P. Kearney and H. Zhang. "An Information-Based Sequence Distance and Its Application to Whole Mitochondrial Genome Phylogeny." *Bioinformatics* 17, no. 2 (2001): 149-54.

M. Li, X. Chen, X. Li, B. Ma, P. Vitányi. "The Similarity Metric." In *Proc. of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003, 863-72.

J. McAllister. "Algorithmic Randomness in Empirical Data." *Studies in History and Philosophy of Science Part A*, 34, no. 3 (2003): 633-46.

D. Parrochia. "L'idée d'une information-limite." *Philosophie du langage et informatique.* Hermès, 1996.

H. Leff and A.F. Rex, eds. *Maxwell's Demon 2: Entropy, Classical and Quantum Information, Computing*. Taylor & Francis, 2002.

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

J.G. Roederer. *Information and Its Role in Nature.* Springer, 2010.

C. Shannon. "A Mathematical Theory of Communication." *Bell Systems Technical Journal* 27 (1948): 279-423, 623-656.

R.J. Solomonoff. "A Formal Theory of Inductive Inference: Parts 1 and 2. *Information and Control* 7 (1964): 1-22 and 224-254.

P. Godfrey-Smith and K. Sterelny. "Biological Information." The Stanford Encyclopedia of Philosophy (Fall 2008 Edition), edited by Edward N. Zalta. http://plato.stanford.edu/archives/fall2008/entries/information-biological.

K. Svozil. *Randomness & Undecidability in Physics*. World Scientific, 1994.

J.A. Wheeler. "Information, Physics, Quantum: The Search for Links." In *Complexity, Entropy, and the Physics of Information*, edited by W. Zurek. Addison-Wesley, 1990.

S. Wolfram. *A New Kind of Science.* Wolfram Media, 2002.

H. Zenil and J.P. Delahaye. "On the Algorithmic Nature of the World." In *Information and Computation*, edited by Gordana Dodig-Crnkovic and Mark Burgin. World Scientific, 2010.

H. Zenil, J.-P. Delahaye, and C. Gaucherel. "Image Information Content Characterization and Classification by Physical Complexity." arXiv:1006.0051v4 [cs.CC], forthcoming in the journal *Complexity*.

A.K. Zvonkin, L. A. Levin. "The Complexity of Finite Objects and the Development of the Concepts of Information and Randomness by Means of the Theory of Algorithms." *Jour Uspekhi Mat. Nauk* 25, no. 6 (1970): 85-127.

Too Much Unity: A Reply to Shanahan

Pentti O A Haikonen
University of Illinois at Springfield

I wish to thank Murray Shanahan for kindly responding to my modest article, where I highlighted some problem areas in Global Workspace models (Haikonen 2011). Shanahan's expert reply clarifies many points, but certain issues remain.

My main worry was related to the requirement of a common code that is present in Blackboard derived systems. Shanahan assures that in his model no common code is required. I can accept this, provided that the modules communicate with each other in an associative or similar way. This question may be considered solved, but another fundamental issue remains.

Is unity a fundamental property of the conscious condition? Does the conscious condition necessitate the coalition of all processes? Shanahan seems to think so and writes in his reply: "But a fundamental property of the conscious condition, whether or not it is realized in a biological substrate, is unity." Shanahan explains how this is achieved in his model: "Only one coalition of processes at a time can triumph and dominate the dynamics of the system. In the conscious condition, the brain and the multitude of processes that constitute it act as an integrated whole" (Shanahan 2011). Shanahan laments that "The difficulty here (in the Haikonen architecture) is that there is nothing to prevent competing coalitions of processes (modules) from simultaneously forming while 'focusing their attention' on different objects." Shanahan is correct here, the Haikonen architecture does allow several separate simultaneous coalitions of processes, while the Shanahan architecture allows only one coalition of processes at each moment.

How integrated is the conscious condition? This is a philosophical and practical question. There must be some integration and cross-connections between different modules, so that the requirements of internal (and external) reportability and short-term memory making can be satisfied. Does this mean that all the cross-connections between the various modules should carry only those signals that are related to the consciously perceived event? In the Shanahan (2010) model this seems to be so; the limited bandwidth communications infrastructure allows only one coalition of processes at a time.

However, in addition to the ongoing conscious experience, conscious humans are able to process additional semiconscious or non-conscious tasks at the same time. These tasks are not only some limited sub-conscious processes within the various modules, no, there are tasks that call for limited coalitions of modules. When I write this reply, I am also aware of my surroundings even though my mind is focused on the topic of this text. At the same time I am also able to use the keyboard and mouse; these acts call for certain sensorimotor coalitions. These coalitions are separate from those that generate my thoughts. The "bed time story effect" (Haikonen 2003) is another example. Here a parent reads a story to a kid, while consciously thinking completely different thoughts. Obviously the story reading calls for the coalition and cross-connection of visual and speech producing modules, but this limited coalition will not be able to form short-term memories, so the reader will not be able to report what has been read. The story reading has not been fully conscious. "Automatic" non-conscious and semiconscious actions do call for the coalition of various resources and these cannot be formed if the cross-connection lines of the communications infrastructure are reserved for other use.

The Haikonen (2003, 2007) architecture allows simultaneous coalitions, of which the ones that are able to produce reports and short-term memories, would be considered conscious in afterthought (provided that the phenomenal aspects would be satisfied; that would be another story). This property of the Haikonen architecture is not a shortcoming; instead, it is a designed and planned feature that allows the maximum use of the cognitive resources of the system as required at each moment. This feature is denied in the Shanahan architecture. Therefore, I suspect that the Shanahan architecture would not be an optimum model for a robot brain or a proper model for the human brain, which is able to support simultaneous conscious, semiconscious, and non-conscious processes and actions. The limited bandwidth Global Workspace communications structure is not only redundant, it is a limiting factor.

References

Haikonen, P.O. *The Cognitive Approach to Conscious Machines*. UK: Imprint Academic, 2003.

Haikonen, P.O. *Robot Brains, Circuits and Systems for Conscious Machines*. UK: Wiley, 2007.

Haikonen, P.O. "Flawed Workspaces?" *APA Newsletter on Philosophy and Computers* 10, no. 2 (2011): 2-4.

Shanahan, M. *Embodiment and the Inner Life*. Oxford: Oxford University Press, 2010.

Shanahan, M. "Unity from Multiplicity: A Reply to Haikonen." *APA Newsletter on Philosophy and Computers* 10, no. 2 (2011): 4.

Philosophy and Online Education

Reflecting Back Twenty Years

Ron Barnette
Valdosta State University

Long before today's electronic learning environment was even imagined by many philosophy instructors, I had a somewhat unusual idea in 1991—which some called crazy and even unprofessional, but not my colleagues at the university, I proudly add—**to teach a full-credit philosophy class totally online.** As a professor who had been teaching successfully since 1972 at Valdosta State University in Georgia, and as department head since 1974, I proposed to enlist the Internet (not generally called that then) to assist with online instruction in some fashion. That moment became the onslaught of my early voyage into an e-learning experience that continued throughout the 1990s and beyond. Please allow me to reflect on the original plan twenty years later and share some salient experiences in the context of this historical perspective.

An early form of email connection was afoot, and I realized that this medium could well serve as an asynchronous linkage to students anywhere, anytime; all with dialup connectivity, for those who had access, obviously, as there were no ubiquitous high-speed networks back then…at least not readily available. Further, I knew that I had to, somehow, arrange for online reading materials for online access, if this course plan was to work. I was also convinced that connecting to students broadly was a concept much in keeping with my own commitment to reaching students beyond my physical setting, in order for them to make virtual contacts with those students at the university. But could this plan be realized, and how? There was at that time no university department for online support, after all, so the reality of forging a bootstrap operation was significant in this experimental era.

Fortunately, I had a very savvy former student, Joe Newton, who was working with the university in what would be not much later referred to as "Information Technology." (Mr. Newton is now the University's CIO of IT, after all these years, I must add proudly.) Joe and I created a Philosophy online presence with Valdosta State's **Philosophy Gopher Server**, at (looking back now) the awkward address of gopher://catfish.valdosta.peachnet.edu/11/ccr/subjv/phi!!

The **Gopher protocol** was a TCP/IP Application layer protocol designed for distributing, searching, and retrieving documents over the Internet. Strongly oriented towards a text menu-document design, the Gopher protocol was a predecessor of the World Wide Web.

In addition to a few other items, we managed to include some 100 full texts of Philosophy for the online community and for my online project! These would be my library for reading materials for anyone, anywhere, after all. Our Valdosta Philosophy Gopher was launched in 1992 with much success as it turned out, as world-wide Gopher servers began to multiply and to reference the Valdosta Philosophy Gopher and our text archives. Mind you, at this time there was not an online, dedicated Philosophy resource of this sort other than ours at Valdosta, a fact that I suspect many 2011 students would find utterly incredulous. But we pioneers were in cyberspace heaven! I immediately created and conducted a trial course online in 1992 with our Gopher presence, named "Problems in Philosophy."

By the summer of 1994, after much reflection, discussion, and experimentation about how a world-wide online Philosophy course could be conducted entirely through asynchronous email means, with textual support through our Philosophy Gopher Library, I decided to launch and conduct the first offering of **PHICYBER, A Virtual Classroom: the Electronic Agora.** Philosophy in Cyberspace was born, and here's a brief announcement from the *Chronicle of Higher Education* which mentioned this and which I received:

"August 10, 1994

Philosopher Creates On-line Classroom

A professor of philosophy is teaching a course this summer without a classroom and without any of the participants' meeting face to face. Ron Barnette, chairman of the philosophy department at Valdosta State University, is running "A Virtual Classroom: the Electronic Agora" entirely over the Internet. Discussion topics for a week are posted on a mailing list to which all members of the class subscribe. The list enables a student's response to be circulated via e-mail to everyone in the class. Readings are done using more than 100 texts that are available on a Gopher server maintained by Mr. Barnette's department.

The professor says that eliminating the need for physical presence and synchronous discussion has resulted in better work from his students. Conducting a class dialogue via e-mail allows all the students to develop their thoughts, he says. Being on line, Mr. Barnette says, insures that reticent students air their ideas and get responses. "To that extent, the level of sophistication and reasoning that all the students develop is much greater than in a regular class setting," he says."[1]

This early experiment was surprisingly enriching, as I was to later write in this message in 1999 for students in my PHICYBER course, which I quote:

In the so-called Marketplace of Ideas, a standard metaphor for depicting philosophical dialogue and participants' give-and-take, it is generally assumed that an open, critical thrashing-out of viewpoints enhances the pursuit of truth and the level of communication between the parties. Yet we all realize how personal bias, subjective presumptions (conscious and unconscious), based largely on another's physical presence (age, race, gender, demeanor, body language, or what have you), affects the tone and direction of such dialogue. In short, we all tend to bring to discussion with others attitudes based on how we perceive one another. What would happen if a group of interactive university classroom participants met and engaged in dialogue without each others' physical presence, only through personal identification based on computer exchanges? How would only the domain of mental thoughts, as represented through electronic dialogue, affect the dialectical development in this new cyberspace version of the marketplace of ideas, where one's ideas and expressed reasons are the class members' only points of contact? Indeed, how would we relate to each other, where "You are what you write" is the criterion for cyberspace personal identity?

In part to explore these questions, a full-credit Philosophy special topics course was developed and offered at Valdosta State University during summer quarter, 1994—probably the first such philosophy course at any university. "PHI 4800—PHICYBER, A Virtual Classroom: the Electronic Agora" transcends traditional space/time constraints on educational settings. It originally included local members—both on and off campus—as well as participants from other states and countries, thus offering a wide diversity of representation. Class members are virtually present around the clock, for the weeks during the class. An expanded version was conducted summer, 1995, and included 71 participants from nine countries... the Electronic Agora indeed went international! In summer, 1996, 111 participants joined PHICYBER for the global classroom, from 11 nations and 5 continents! It is apparent that the world marketplace of ideas is now a reality online...And during the 1997, 1998 and 1999 summer sessions, the international electronic dialectic continued, with over 110 participants joining the agora, from five continents, discussing topics of multinational significance. What a classroom in cyberspace! Through an academic list named PHICYBER (Philosophy in Cyberspace), all class discussion, debate, and critical thinking exchanges are ongoing, as relationships develop, and group affinities evolve, all driven by the diverse thoughts and experiences exchanged through the e-mail Internet conveyance, the electronic infrastructure of this technological marketplace of ideas, or electronic Agora. Weekly assignments in the Virtual Library comprise the research and 'paper writing' aspect of the class, with major philosophical works made available on-line, and each assignment is designed to enhance access to resources and Electronic Texts, highlighted internationally by the American Philosophical Association through the Internet. We also feature my creative and fun Zeno's Coffeehouse (est., 1994), for relaxation and brain-exercises!

As the 1990s decade evolved with PHICYBER, (http://www.valdosta.edu/~rbarnett/phi/phicyber/), then online in 1995 with its web-based presence through earlier browsers Mosaic, Netscape, and eventual Internet Explorer access, the Electronic Agora was finally underway with many participants from around the globe. The worldwide connections with students were amazing, as all this was experimental and unique for each of us in this totally asynchronous learning environment. But how could one do this online with only email connectivity, it was asked by many, including the online learning skeptics? How indeed?

As I then wrote in an article, "Teaching Philosophy in Cyberspace," let me quote from what I remarked:

> An initial challenge that one faces in cyberspace is how to prompt such dialogue. Due in part to the novel forum, we all help to craft who we are through each word, and there are no slips of the tongue that are typically fast forgotten. As one student put it during a media interview about PHICYBER, "It is so different when you have to think through your ideas, put them in writing, and be prepared to back up your views, knowing that once expressed they are out there for the permanent record!" This student alludes here to the fact that all classroom work and discussions are placed in a course archive, and are available to the world for ongoing retrieval and review. One can think of the "in-class" portion of the course as a transcript. There are no voices or accents, no noises, nor distractions based on gender, race, ethnicity or age. Only ideas, and ideas about ideas, formulated, written and re-written, expressed and revisited. In fact, the ongoing discussion in the class *is* the class, and the class *is* the set of ideas expressed. A participant becomes, in a sense, a Platonist in cyberspace, instantiated by material objects and electricity![2]

Looking back at what I wrote, I must say that I shudder over many of the catchy phrases, so arcane nowadays! Yet the language was evolving with the technology, and the 1990s comprised a fast-evolving era into electronic learning, especially as colleges and universities came to realize the power and influence of the Internet on teaching and learning. Faculties nationwide were increasingly encouraged (even pressured) to develop online courses with precious little web-design course support, as this era of institutional online support was still in its infancy. With the advent of "course in a box" technologies, applications such as WebCT and Blackboard took off and were adopted by many of these institutions, yet they often displaced the uniqueness of an individual faculty's class design. But they did standardize and ease the challenges facing institutional faculty support. However, by the end of the decade there was already in place a genuine debate over "course ownership" when it came to class design and delivery. Almost overnight, one had seemingly gone from "bottom-up" faculty experimentation with online learning to "top-down" institutional online structures and course-delivery models. Indeed, the evolution of electronic learning was proceeding with lightning-fast speed, and has never diminished as we now move forth into its third decade.

Relating back to e-learning in the mid 1990s with a reflective glimpse into how the earlier fast-evolving 2000s transpired, I must mention first, Jon Dorbolo and Bill Uzgalis, early fellow pioneers who teach at Oregon State University. Dorbolo's early-90s InterQuest Philosophy course in online learning was a milestone in this era, www.oregonstate.edu/instruct/phl201/, and Uzgalis's web-based course Great Voyages, the History of Western Philosophy from 1492 to 1776, showcased and demonstrated creative course design and access at www.people.oregonstate.edu/~uzgalisw/302/.

And, second, I must mention Peter Boltuc, at the University of Illinois at Springfield, www.uis.edu/philosophy/faculty/research/boltuc.html, who was instrumental in developing in the 2000 e-learning era a full-blown online B.A. degree program, and not just a course. He began with a Philosophy minor online in 2002, which was a catalyst for his seminal work for the B.A. program he designed and which was launched in 2004. In my estimation, this online degree program is still the most thoughtful, supportive one available to date…and there are now several. I claimed this initially in 2006, at a presentation before the North American Computing and Philosophy conference at Rensselaer University, http://www.cogsci.rpi.edu/conferences/cap/schedule.php.

Thank you for the opportunity to reflect a bit on the early days which were exciting for this philosophy e-learning experimenter and electronic classroom architect. As one who helped initiate the effort and who taught some twenty online courses through 2006, perhaps someone just might venture forth with a rich history of e-learning in Philosophy, from Gopher to Web 2.0 and beyond. Wouldn't that be a remarkable story? Meanwhile, I am pleased to have been an earlier party in the e-learning transformation and its continued evolution.

Endnotes

1. "Philosopher Creates On-Line Classroom," *The Chronicle of Higher Education*, August 10, 1994. Washington, DC.
2. Ron Barnette, "Teaching Philosophy in Cyberspace," *The Digital Phoenix: How Computers are Changing Philosophy*, edited by Terrell Ward Bynum and James H. Moor (Blackwell Publication, 1998), 325.

Reflections from Teaching Philosophy Online

Frank McCluskey
Scholar in Residence, The American Public University System

It has always been a conceit of mine, that philosophers should be a shade smarter than most other people in the academy. Notice I said "should be." It is in our DNA not to assume things we cannot prove. It is our business to reject those things about which we are not absolutely clear. It is our curse to question those things everyone else accepts as bedrock certainties. Other disciplines are built on mountains of presuppositions and questionable leaps while we have the luxury of returning to the beginning and to ask how we got where we are. The good news is that philosophers should constantly be learning. The bad news is that philosophers are constantly learning how much they do not know. The wisdom of Socrates was a simple proposition. He alone was aware of all that he did not know for certain.

By the late 1990s I had been teaching philosophy for more than a decade. Being a professor at Mercy College in New York, a small liberal arts college, I spent a good deal of my time teaching an Introduction to Philosophy taught in historical way beginning with The Apology of Socrates. Outside of the classroom I had been interested in linguistics and epistemology. In the 80s I had been playing with artificial intelligence languages such as Turbo Prolog and LISP when I came across what we called in those days "electronic bulletin board systems" or BBSs.

All of the words used in cyberspace had to be borrowed from the physical world and brought with them the connotations that they have. The metaphor of a bulletin board was interesting. Cork bulletin boards used to be a very cheap and democratic way to buy and sell things, to look for a job or find a roommate. You would fill out a piece of paper with your offer and write your phone number numerous times and then clip them so that people could tear off that number and call you later. This meant that you didn't need to be present for the sale to be noticed. In the absence of the seller, the process could go forward. People could read and react at their leisure. Today we would call this "asynchronous."

When I first learned about BBSs it brought several questions to mind. How could this augment the classroom? How would this change the conversation? Could a philosophy course be taught at all online? I decided to experiment by asking a number of students in my on-campus philosophy class to dial in at night for a little extra credit. In the beginning I ran a BBS out of my apartment by unplugging my phone line at night and plugging it into my modem. Only one student could call in at a time so I gave them a half hour each and they would dial in at an assigned time block. The technology was a challenge, the lines were slow, and there were technical problems every night. But for all of the challenges, there was something interesting going on. The "conversations" that we were able to have at night showed the possibilities of the new medium.

With the success of my experiment with my philosophy class I convinced a number of other colleagues in different disciplines to see how it would work with their classes. In the beginning there was a small group of faculty who taught composition, business, political science, history, music appreciation, math, and biology. Without a clear idea of where we were going we began to play with BBSs and then talk about what it meant for our teaching. Decades later I heard song lyrics that reminded me of how we were teaching online in the 80s.

"Where were they going, without ever knowing…the way."

Every discipline is different. Teaching math is different than teaching music appreciation. Teaching biology is different than teaching French. One of the differences is the amount of content that each of the disciplines needs to get across and the number of assumptions each discipline has.

Like many professors of philosophy who teach sections of Introduction to Philosophy, I had developed two different lives. In one life I was a teacher and my aim was to excite, entertain, and wake up my students from their dogmatic slumbers. In another life I was a solitary scholar reading Hegel, wrestling with logical issues, and working with textual problems on my own. In the days before the real power of the Internet, many of us found ourselves as the only professor of Greek Philosophy or Logic or German Romanticism in our universities. It was a lonely life for me that would not last forever. Both my teaching life and my scholarly life would be transformed by the online world in ways I could not have envisioned.

I can still recall the excitement of a small group of faculty who taught different subjects that came together at a local pub one night to see how we could leverage this new medium in our teaching. As a philosopher I have recognized the key role that pubs have played in the history of our discipline. It seemed like an impossible dream and more than once it seemed the technical difficulties would sink the whole project. The good news was this was a faculty driven initiative without any grant or administrative funding, support, or pressure. We were free to play and try new things and fail. It was more of a pedagogical and philosophical experiment than a project with a goal. Many years later as I became familiar with the growth of online learning, I found that there are two major ways that online learning comes into a university. In one way some president, provost, or board member decides it makes economic sense and it is hoisted on the faculty from above. In other schools, online learning bubbles up from the faculty. Usually there is a champion or champions who are doing it for the sheer love of teaching. In the second case, online learning has a different feel because the faculty owns it and they feel a genuine concern in its care and feeding.

In the decades that have followed that heady meeting where we dreamed of possibilities and ideas, I have learned many lessons. I would like to present a few ideas that have come to me in that time.

1. Philosophy is more suited to online education than many other subjects.

When a dedicated group of faculty at my college began playing with online learning in the late eighties, I quickly became aware of the difficulties other professors faced heading into cyberspace. Music appreciation had the difficult task of getting copyright permission for clips and then getting them to work online. In those days it was difficult to get math symbols online so math and science professors had work to do. Courses in which there were long lectures and loads of content had to put up long and boring "pages" that would get the content across. I could not have guessed then how long it would take to get whole science labs up and online or how Citrix would be needed to compile computer languages over the network. There were all sorts of issues about symbol sets, sound bites, and links that caused our little band endless consternation. Philosophy went online amazingly easy and seemed absolutely fitted for online learning.

Some of this may have had to do with my teaching philosophy and style. Rather than assigning huge blocks of text I would assign just a few pages to read with the critical philosophical eye that would dig deep in one spot. I didn't need my students to know all the writings of Aristotle. But I did want them to wrestle with some key passages that would give them a sense of what he trying to do and at the same time sharpen their critical abilities. I was always amazed that as many times as I taught Plato's Apology of Socrates, I never got bored or fell into some clichéd lecture I would give over and over. The richness of the questions about what is wisdom, what is the business of philosophy, and what is the relation of philosophy to politics and the arts seemed to be an endless gold mine out of which I would bring up a new treasure every semester. At the same time the text of Plato's Apology was so apparently simple that many undergraduates can jump in and begin to learn from the arguments. Asking students to argue the big questions while using support from the text made the class open a torrent of messages. From my first class online I saw that the exchange of arguments and the dialogical nature of the philosophical conversation made it ideal for the online space.

2. Philosophy online gives more students the opportunity to philosophize.

As a student myself I attended many college classes where I took notes silently and waited anxiously for the exams and term papers to be graded so that the mystery of my grade would be revealed. I once read that philosophy is the course that students remember the most when they graduate no matter what their major is. Student who come back to talk to me years after graduation often still recall questions such as, "How do you know this chair exists when no one is in the room looking at it?" "Is lying ever ethical?" or "How do you know this is not a dream?" Before the advent of the online classroom I would select an adventurous student in the front row, who had a pretty healthy ego, and let them do a lot of heavy lifting for the class on questions like this. But in the online classroom there is the time and space for a more widespread discussion and a greater involvement of classmates. It is well known that introverts prefer writing to speaking. So here is medium where they are able to take chances in ways they might not in the bricks and mortar classroom. Philosophizing is best done one on one so questions can be addressed and the conversation steered towards the student's particular interests. This medium allows for that in ways that are impossible in a bricks and mortar classroom.

3. Philosophy calls for the student to make the argument their own.

One of the first differences I noticed when I went online was that the students continued the conversation when I was absent from the class. When I taught in the bricks and mortar classroom I would frequently go out after class to a coffee house or tavern with a few of my most enthusiastic students and continue the conversations. But when the class was over the vast majority of students packed up and went back to their dorm rooms or homes. In the online classroom, however, I would sometimes log back on in the morning to find dozens of messages exchanged between a wide number of students and whole new threads of conversation begun. It was a class outside the class that was running on its own power. There are disadvantages to this as a wrong turn can lead someone a long way before the instructor checks back in. The good news is that the class became a learning community where the teacher was less important than the lessons. I recalled my first visit to Oxford when I sat in a pub at night and listened in on conversations on Joyce, Uncertainty, and Proust all without the guidance of a professor. When I launched my first virtual class I gave my dispersed students the chance to sit together and philosophize on their own.

4. I learned where my teaching succeeded and needed work.

Unlike the physical classroom the digital classroom leaves a record of every interaction. This means there are quantitative indicators of where students react and where there is less classroom energy. Early on I was able to quantify the number of posts to discussion boards. Later I found tools that could analyze the qualitative elements in student work. I could create a heat map much like the weathermen do on TV and see where student involvement was active and where it was lagging. I could see where the posts were all aimed at the instructor and other subjects where peer-to-peer learning took place. This would allow me to adjust the questions, shorten or lengthen the readings, and change my elucidation of the texts. For the first time I could see, color-coded like a traffic light, where the learning was speeding up or slowing down. The Greeks made a distinction between knowledge and opinion. Before I had a digital map of the class it was my opinion about how successful my teaching was. Once I had the data, I had knowledge.

5. It opened up a new world to philosophize about.

Whatever "cyberspace" is (and after two decades I am not quite sure), it is fascinating. The relation between speaking and writing has been a subject debated by philosophers since Plato in his Phaedrus wrote that writing is further removed from reality than speaking. In the Phaedrus, Plato argues that something is lost in writing that is present in speaking. Philosophizing about speech continues in our time with thinkers as different as Wittgenstein, Heidegger, and Derrida, to mention only a few. A whole bank of new questions arose when the classroom went virtual. Once the classroom went online is it still a "classroom"? Is a webpage still a "page"? What does it mean to "talk" online? How is our online persona different from the one we speak from in front of another? These and other questions have moved me to rethink almost everything I believed about education. The dynamics of the online classroom are so different that it has changed how I thought about teaching, time, and space.

6. Philosophers can learn a lot by teaching online.

Since online learning has emerged all sorts of questions have arisen because for the first time we have something to compare the classroom to. There are questions about are students cheating online. As someone who has taught symbolic logic for decades I can tell you a lot of my students in the bricks and mortar classroom tried to get a little help with their in class final exam. There are questions about if online learning is "real learning." This question is the most interesting to me and it takes me back to Plato. In Book VII of his Republic, Plato gives us the allegory of the cave. This parable tells of a man who is held prisoner in a cave for his entire life. During this time he and his other prisoners come to mistake the shadows and echoes of the cave for real things. Because they never had anything to compare them to they all agree that this reality is the only truth. But one prisoner escapes from the cave and goes up into the stars and eventually sees the light of the sun. He now has something to compare the cave with. It is said that when he returns to the cave to free his fellow prisoners his vision in the cave was poor as he was no longer accustomed to the dark. He begins to tell his fellow prisoners a tale that seems so unlikely that he puts himself in danger.

Since the Founding of the University of Bologna in the last years of the eleventh century till just a few decades ago there was one model for the classroom and all of us came through that system. With the birth of the online classroom we now have something we can compare with the bricks and mortar classroom. For the first time we can ask questions that before may have seemed impossible.

When Plato's prisoner returns to the cave, he comes back with a new wisdom, a new outlook, and new experiences. Those in the cave can learn a lot if they are not afraid of something new and different.

Select Bibliography

Anderson, Terry. *The Theory and Practice of Online Learning*, 2nd ed. Edmonton: Athabasca University Press, 2008.

Beck, Evelyn, and Grieve, Donald. *Going the Distance, A Handbook for Part-Time & Adjunct Faculty Who Teach Online*, 1st ed. Ann Arbor: Adjunct Advocate Inc., 2005.

Bonk, Curtis J., and Zhang, Ke. *Empowering Online Learning: 100+ Activities for Reading, Reflecting, Displaying and Doing*. San Francisco: John Wiley & Sons, 2008.

Conrad, Rita-Marie. *Engaging the Online Learner*. San Francisco: John Wiley & Sons, 2004.

Elbaum, Bonnie, et al. *Essential Elements: Prepare, Design & Teach Your Online Course*. Madison: Atwood Publishing, 2002.

Kemerling, Garth. "Teaching Philosophy on the Internet." 20th World Congress of Philosophy, Boston, 1996. http://www.bu.edu/wcp/Papers/Teac/TeacKeme.htm

Pallof, Rena M. and Pratt, Keith. *Assessing the Online Learner: Resources and Strategies for Faculty* (Online Teaching and Learning Series). San Francisco: Jossey-Bass, 2009.

Presksill, Stephen. *Discussion as a Way of Teaching*. San Francisco: John Wiley & Sons, 2005.

Smith, Robin M. *Conquering the Content: A Step-by-Step Guide to Online Course Design* (Online Teaching and Learning Series). San Francisco: Jossey-Bass, 2009.

A Comparison of Four Distance Education Models

Terry Weldin-Frisch
University of Illinois

Abstract

This paper explores, from the perspective of the author in the role of distance learner, four elements deemed critical for the delivery of a quality distance educational experience (Faculty Contact, Course Requirements, Student Services, and Learning Resources). In the early stages of the author's distance educational experience a naïve assumption was made that most institutions of higher learning engaged in the delivery of distance education programs, if accredited, would generally deliver roughly equivalent educational experiences. After completing coursework at three institutions delivering distance education and going through the registration process at a fourth institution, the author has come to the conclusion that not all institutions do in fact deliver roughly equivalent educational experiences when their programs are assessed using these critical elements as filtering criteria. Thus, it seems to the author that if there is room for improvement, then institutions delivering distance education programs must engage in a process of making substantive, ongoing efforts to assess and improve the distance learner's experience of each element here discussed. If this sort of substantive, ongoing assessment and improvement process is carried out on a regular basis, it seems a fair conclusion that the quality of the distance learner's educational experience can be expected to improve commensurately.

Introduction

Upon entering into computer assisted distance education, the author assumed that institutions engaged in the delivery of distance education programs generally deliver roughly equivalent educational services to students. It turns out that this assumption was somewhat naïve when comparing four critical elements relating directly to the students' online educational experience. Some of those experiences were positive while others made the journey more challenging. However, even with these challenges, the author believes that distance education is an effective method of delivering philosophical content.

This paper discusses the author's experience as a student taking online courses in philosophy at three educational institutions. In addition to the three institutions where coursework was completed, this paper includes an assessment of one additional educational institution the author did not attend but where he was accepted into its online MA in Philosophy program. Two of the institutions discussed in this paper are in the United States and two in Europe.

Assessment and discussion of four critical elements

In Tables 1 and 2 that follow, the four institutions being assessed are Trinity College and Seminary (TCS), Newburgh, IN; University of Illinois at Springfield (UIS), Springfield, IL; University of Oxford, (UO), Oxford, England; University of Wales, Trinity St. David (UW), Ceredigion, Wales. These institutions are assessed based on the author's specific experience of four critical elements. These elements relate to attending the institution as a student or, in the case of the one not attended (UW), focus on the experience related to becoming familiar with the organization via the UW website, in consultation with the MA in Philosophical Studies program coordinator and through the application process. Table 1 analyzes the elements related to Faculty Contact and Course Requirements. Table 2 assesses Student Services and Learning Resources.

Table 1. Faculty Contact and Course Requirements

Elements of Analysis:	1. Institutional Contact			2. Faculty		
	E-mail	Phone	Other	Adviser assigned	Availability of adviser	Faculty credentials appropriate to courses taught
Institution						
TCS	Y	Y	Y[1]	Y[2]	Y	Y
UIS	Y	Y	Y	N	Y	Y
UO	Y	NA	Y	N	N	In some cases
UW	Y	Y	Y	Y	Y	Y

Discussion of Table 1 Elements

Some additional discussion of Table 1 is required to clarify the above assessments. In the case of TCS, the various contact methods and the accessibility of faculty to the student was a function of the student's enrollment in a PhD program for which all support services were provided. The adviser assigned was a function of the department in which the student was enrolled (Department of Philosophy and Apologetics). Further, in the case of the dissertation itself, the student requested a specific professor as his dissertation chairman although the institution did not guarantee that this request would be honored contingent upon a number of variables.

In the case of UIS, while the student had specific instructions for contact of faculty for the course taken, other considerations related to formal faculty access relative to advising was not germane for this student given his non-degree seeking status.[3] While this is true for this student, UIS does place a strong emphasis on student advising at both the undergraduate and graduate level.[4]

The qualification of faculty at UIS for courses taught exceeded expectations. The UIS faculty, although small, boasts professors from some of the top philosophical programs in the world including Princeton and MIT.[5] While taking the online course *Metaphysics & Epistemology* and in personal one-on-one interactions with the faculty the author was and continues to be consistently impressed with the qualifications and professionalism of the philosophy faculty at UIS.

In the case of OU, the author has taken six courses to date through the continuing education department.[6] In addition to the standard method of communication with the tutor through e-mail, one further method of communication is worthy of mention. Generally, with this particular university, the course author and the course tutor are not identical. In the Philosophy of Religion course, the course tutor invited the course author to join the students in a synchronous discussion session at an appointed date and time. There ensued a lively exchange between students and the course author, Dr. Tim Mawson, Tutor in Philosophy at St. Peter's College, University of Oxford, who also happened to be the author of the textbook for the class. This experience stays with the author as one of the more memorable ways of delivering online education. The lively engagement noted in this synchronous event seems to suggest that this methodology could be effectively used for delivery of philosophy seminars at the undergraduate or graduate level. This would necessitate a set of commitments between course instructors and students regarding dates and times when all would come together to discuss weekly reading assignments or other types of assignments such as initial discussion of papers being written by students for the class or for publication, etc. Nonetheless, this seems a quite reasonable approach to mitigation of some of the current arguments against distance or online education such as the loss associated with not being together in the same room. Further, with technology such as Skype, these arguments become less convincing. There is simply no good reason to think that this particular educational delivery method cannot be perfected in such a way that long held predispositions regarding the second-rate quality of online philosophical education need continue to dominate a twenty-first century educational milieu.

It may be noted that the column titled "Credentials appropriate to courses taught" for the OU faculty states this to be so "in some cases." That means, in some cases the tutors are philosophy PhDs who are teaching and publishing in their areas of specialization but in other cases they are PhD candidates finishing their dissertations. So, although the tutoring has been excellent in every course, in the author's opinion, in some cases they are not PhDs working in the field in the sense of teaching and publishing.

In the case of UW**,** where the author went through the entire procedure of contacting the institution, applying for and being accepted into the MA program, the process itself was judged to be clear and uncomplicated from start to finish. Contact was made easy as all initial communications flowed directly through the appropriate program advisor for the particular area of specialization a student is interested in pursuing. So, although the author cannot comment on the educational experience itself, the experience of applying to become a student was positive. Thus, one would like to think that the actual educational experience itself would follow a similar user-friendly trajectory.

Table 2. Student Services and Learning Resources

Elements of Analysis:	3. Course Requirements			Student Services		
	Syllabus[7]	Methods of evaluation[8]	Overall Clarity of Expectations[9]	On-campus Library	E-resources	Timeliness
Institution						
TCS	Y	Y	In most cases	N	X	Y
UIS	Y	Y	Y	Y	Y	Y
UO	Y	Y	Y	Y	Y	Y
UW	NA	NA	NA	Y	Y	Y

Discussion of Table 2 Elements

As with Table 1, Table 2 requires some additional explanation. TCS, UIS, and OU all provided syllabi for courses offered.[10] In the case of TCS, there was one syllabus for each course containing undergraduate, graduate, and doctoral requirements. This is what TCS describes as a modular format. Within that one syllabus the course purpose and course objectives were differentiated for each degree level.

The course objectives, categorized according to Bloom's Taxonomy, were used by TCS faculty to distinguish learning outcomes for the different degree levels within the modular format. So, as an example, for the History of Modern Philosophy course an undergraduate student was directed to be familiar with objectives 1-4, a master's student with objectives 3-5, and a doctoral student with objectives 4-6. In objectives 4-6, the doctoral student was working on developing analytical, critical, and evaluative skills aimed at "expanding the student's ability to check one's worldview commitments for comprehensiveness, coherence, cohesiveness, adequacy, and applicability."

TCS provided two primary evaluation methods for each course: research papers and summative exams. At the doctoral level this criteria typically translated into three research papers per course in the 20-25 page range. At the end of each course, the student was required to take one summative examination to demonstrate mastery of the course material.

In terms of overall clarity of course expectations TCS did a good job with the exception of communicating the importance of taking the Research Methods course early in the program.[11] This is critical from the author's perspective because it would have changed how he approached each course and the papers he would have chosen to write. Unfortunately, given the fact that he took the Research Methods course toward the end of the program where he was advised that it would have been better to take it earlier in the program by the course instructor, it made the dissertation project more challenging than it needed to be.

Another point needs to be made regarding the methods of evaluation utilized by the various organizations under consideration here. In the case of UIS and UO, there was opportunity to interact with the professors and other students by means of asynchronous technologies such as Blackboard and Moodle. This adds a dimension that was missing from TCS at the time the author was completing the PhD program. Although this aspect of online educational delivery has since been improved by TCS, at that time it made the overall educational experience seem more isolating than the asynchronous models being utilized at UIS and OU. In a recent discussion with the author's dissertation chairman it was pointed out that at the time the author was working on his PhD the online educational technology was in its infancy and thus had not yet been adopted by TCS. However, it is also important to point out that even at that time there was technology available for the delivery of online education. Dr. Piotr Boltuc from UIS pointed out that UIS was already delivering online education via a program called Web Board as early as 1998. Of course, it is also important to note in this context that UIS has been an innovative leader in the delivery of online education since the 1990s. Even today, there appear to be those who continue to resist this important paradigmatic shift in education. The picture here reminds one of those who continue to practice normal science long after the scientific revolution has shifted to a new paradigm.

It seems almost too obvious to say that direct access to a substantive library collection is essential for the online student. A comparison between TCS, UIS, and OU's provision of this resource is informative. In the case of TCS, at the time the author was completing his PhD, students could borrow books through an arrangement with The Congregational Library in Boston, MA. However, this meant having to do the research at a distance, then order the book by e-mail and wait for the book to be shipped from the east coast to Illinois. In addition there were some electronic databases available but these were limited compared to what is available through both UIS and OU (online and on campus). For the author, two strategies made the research problem more manageable. First, there was a seminary within thirty-five miles of Springfield, IL, and they agreed to allow the author to use their collections and to check materials out of the library. Second, in 2004 the author made a connection with the chairman of the philosophy department at UIS and was granted status as a pro bono teaching assistant. That opportunity opened up the UIS library system as well as the I-Share catalog[12] which made research more efficient and comprehensive than it otherwise would have been. In fact, this access made it possible for the author to complete his dissertation which otherwise might have stalled out due to the unwieldy system then in place at TCS.[13]

Finally, timeliness in Table 2 is understood to mean the length of time a student could expect to wait to hear back from someone within the educational organization being contacted. Generally, the communication turnaround time was within the author's acceptable standards in most cases for every educational institution attended.[14] It is important to point out here that many students do not feel the same way. One of the constant comments the author hears from students taking his online courses in philosophy of religion is their appreciation of his promptness in getting back to them. Thus, it appears that this dimension of the quality of online education is important to the student and should be attended to accordingly in order to avoid having students feeling isolated and alone in cyberspace.

Conclusion

In this paper we analyzed four elements identified by the author as critical to his overall educational experience as a distance learner: Contact, Course Requirements, Student Services, and Learning Resources. All four elements are deemed critical to the quality of the educational experience of online learners. First, have a variety of methods available for students to make contact with the institution and with their departmental chair/adviser. This may require thinking about what it would be like if the tables were reversed and the institution or department was the one in the online learner's situation needing answers or information. Second, it is important in addition to contact methods to make sure that online learners know who their advisers are, when they are available, and that they are well-qualified to be teaching the courses they are teaching. A critical point here goes to the issue of availability. Online learners are non-traditional students. Therefore, it is important for them to know that they can make contact with their professors or advisers in non-traditional ways and perhaps at times not necessarily convenient for the professor or adviser. This may require some additional flexibility on the part of the professor or adviser not typically found in the more tradition bound paradigm of office hours typically found on most syllabi. Third, course requirements need to be clearly spelled out. The syllabus needs to be detailed in terms of expectations, due dates, evaluation methods, etc. Online students should be able to answer almost any question they have about the course by simply referring to the syllabus. Fourth, it is important to make sure students have substantive learning resources available to do the work expected in the syllabus. This seems almost tautological. However, it is important that students not be left in a situation where they have to spend substantive amounts of time trying to resolve inadequate resource problems in addition to the work requirements of the course or courses they are taking. It is good to remember that online students are online students for a reason. Typically, it is because they are already very busy with the rest of their life and online education provides them with the opportunity to complete their education while doing other things such as working full-time, raising families, serving their country in the military in far-flung outposts, etc.

Thus, it seems reasonable that educational institutions undertaking the rigorous journey of delivering distance education programs would closely evaluate each of these criteria and make substantive, ongoing efforts to assess and improve the online/distance learner's experience of each element discussed. If this sort of substantive, ongoing assessment and improvement process is carried out regularly, it seems fair to conclude that the quality of the distance learner's educational experience can be expected to improve commensurately.

Endnotes

1. Face-to-face encounters with the professors were available through the TCS Accelerated Studies Seminars offered in major cities across the United States.
2. Three advisors assigned for Ph.D. dissertation. One adviser was an external reader from Northern Illinois University, the committee chairman and third reader were selected from TCS faculty. All three committee members were requested by the student.
3. In this situation, the author was not enrolled as a degree seeking student at UIS and therefore did not have formal access to student advising services provided to degree seeking students. However, the author did have access to philosophy faculty on an informal, ad hoc basis.
4. http://www.uis.edu/uiscatalog/2006_2007_UIS_Catalog/Information/allStudents.html
5. Based on the Overall Rankings of philosophy departments as reported in *The Philosophical Gourmet Report* for 2010 accessed at http://www.philosophicalgourmet.com/overall.asp (accessed July 30, 2011)
6. The courses covered so far are Epistemology, Metaphysics, Moral Philosophy, Philosophy of Mind, Philosophy of Religion, and Philosophy of Science.
7. Is there a syllabus for the courses?
8. Is there a variety of methods of evaluation for the required course work?
9. Is it easy for the student to understand exactly what is expected in order to receive credit for the course?
10. Although the student did not attend UW, it was noted that in some cases the syllabus for certain graduate philosophy courses was provided on the philosophy program's website.
11. This explains the "In some cases" comment found in Table 2 under "Overall clarity of expectations."
12. I-Share includes the resources of 76 Illinois libraries belonging to CARLI, the Consortium of Academic & Research Libraries in Illinois.
13. It should be noted that since the time the author completed his program at TCS they have made many substantive changes and have many more resources in place than they did at that time. For more information on what is available at TCS, the reader may wish to visit http://lessons.trinitysem.edu/course/view.php?id=2234.
14. This was also true of UW although the author did not attend. However, the response times during the early contact and application phase were excellent overall with one or two exceptions.

An Invitation for Reflection: Teaching Philosophy Online

Kristen Zbikowski
Hibbing Community College

Introduction

While attending the 2011 Central Division APA conference, I found myself engaged in several casual discussions with colleagues. If it was mentioned that I was going to be taking part in the Committee for Computers and Philosophy's discussion of online education in philosophy, there would sometimes come a rather curious moment in the conversation. I began to think of it as the "microwave question moment." Many years ago, when microwave ovens first were advertised, there was a rather clever ad depicting a salesperson standing in front of a crowd of people, going on and on about all the wonderful advantages of microwave cooking, until a deep voice from the back spoke up with: "But does it brown the food?"

Online course delivery has become increasingly prevalent in higher education, and there is often great pressure on educators to deliver courses in this manner. There is, however, still quite a bit of skepticism from those who have not ventured into online teaching. They ask the educational equivalent of the microwave question. "Can students learn well, online? Can you teach philosophy effectively, online? And finally, most importantly, does the online venue serve student interest?" My answer is: "Yes. It browns the food!" Online content delivery, when done skillfully, has some great advantages for the student, and may be especially well suited for the teaching of philosophy.

I began my teaching career in the traditional face-to-face classroom. I did not dip into online teaching until it was suggested to me that it would increase my employability if I "volunteered" to teach online as well. I started out rather skeptical of the ability of online delivery methods to meet student learning needs. I had heard negative things about online courses; that students felt abandoned by the instructors, that it was confusing and hard to stay on schedule, and that there was none of the human contact which helps students understand difficult topics.

Obstacles to Learning

It was a pleasant surprise, therefore, to find that online learning can surmount these obstacles beautifully, and even give students certain advantages that they may not get with face-to-face instruction. Learning, beyond the ability to retain facts and recite them by rote, takes time and effort. The learning taxonomy that Benjamin Bloom and colleagues developed back in 1956 seems an especially appropriate illustration of the type of learning that the study of philosophy demands; taking in new ideas, understanding them, applying them, being able to integrate them into one's worldview, and, finally, gaining the perspective to evaluate them critically.[1]

In order to do this, students must first absorb the ideas, through reading text, instructor commentary, or being given a topic to research independently. Then, they must engage with the ideas, to ensure that they have a good understanding of the new material and can apply it appropriately. After that, the student can figure out what William James might call the "cash value" of the ideas as they fit in with, or clash with his or her worldview and values.

In a perfect world, students and educators would buckle down to the work of learning, with no further complications beyond surmounting different learning styles and levels of readiness. In actuality, there are additional obstacles that must be overcome. The study of philosophy is much more concept driven than many other disciplines. This can cause problems for some students simply because they are not used to engaging in the deeper levels of critical thinking, synthesis and analysis. They struggle, and must be guided into learning these skills.

Within the study of philosophy the very topic material can be rather daunting for some students because what they learn may clash with their current beliefs and throw them into confusion. For some students, venturing to explore topics of moral philosophy or philosophy of mind may be an act of bravery. Students may be reluctant to fully engage with the course material because of worries about what they may encounter.

The next obstacle to be overcome lies in the fact that learning is best done in a social context. According to Cozolino and Spokay (2006), learning is much more of a socially dependant and interactive material than the stereotype of the "lone genius" would have one believe. They write:

> The brain is a social organ innately designed to learn through shared experiences. Throughout the life span, we all need others who show interest in us, help us feel safe and encourage our understanding of the world around us. Brains grow best in this context of interactive discovery and through cocreation of stories that shape and support memories of what is being learned.[2]

We learn more easily when we not only engage with new ideas, but when we do so by interacting with others. This can be a barrier for some students because they may be shy about speaking in a classroom setting, or worried that they will set themselves up for ridicule or argument by speaking their beliefs.

Finally, what I call "outside interference" can also be an obstacle to learning. Students may be juggling conflicting time commitments, have problems in their personal lives, be prone to procrastination, or just be uncommitted to school or the particular course. In these cases, it may be difficult for the student to put in the sustained time and effort that proper learning requires.

Online Course Delivery Strategies

Online course delivery, when done well, can address all of these obstacles. I have been experimenting with various online teaching strategies and I have found some practices to be enormously effective. To begin, I have found that in both face-to-face courses and online courses, the instructor presence can make or break the course from the outset. A strong instructor presence and good classroom control allows students to relax, knowing that the social interactions in the classroom will be respectful, and incivility will be promptly addressed.

An interesting study by Credence Baker published in 2010 provides detailed information about instructor presence and immediacy in the virtual classroom. Baker has found instructor presence in the online setting to be a statistically significant predictor for student learning, cognition, and motivation.[3] To establish and maintain a strong presence in the online classroom, I make it a point to interact with each student at least once each week, and answer questions, e-mails, and pages promptly. In addition to content-related discussion lines, a discussion line dedicated to general questions is always available and tended. Students have multiple ways of communicating promptly with the instructor.

My institution uses the Desire 2 Learn course server, which has a pager feature. This tool is quite similar to instant messaging, and in the first week of class I page students to ask how they are doing and if they need any immediate assistance. Students are encouraged by this interaction and it seems especially vital for those who are new to online learning. I suspect that this initial check-in may be helpful for retaining some new students through the often stressful beginning week of a first online course. Throughout the semester, I continue to page students to alert them of schedule changes, or to "touch base." I have received quite a bit of helpful feedback from students in these interactions, which has allowed me to fine tune my teaching efforts responsively.

Next, to combat confusion over what is expected and when assignments are due, I have found that building a regular routine for course activities is extremely helpful. Students are more likely to participate if they are given a habitual schedule to follow with deadlines built in. Once a routine is established, students are able to plan more effective study and participation patterns. However, trial and error has shown the wisdom of posting deadlines and schedules in multiple places, i.e., a formal schedule, the course home page, and a course calendar, among others.

Time to assimilate, reflect upon, and build associations between concepts becomes especially important in philosophy. The online instructor must bear this in mind when designing the pacing of material to communicate. More content is not always better teaching. There is little point in "overloading" students as the new information will be quickly forgotten if there is no opportunity to integrate it properly within a cognitive framework.

Regular, lighter tasks are much less daunting, and keep students engaged with the course material. Deadlines tend to discourage procrastination. Also, the fact that all assignments are written out helps to further develop both writing skills and critical thinking skills. There has been some discussion in the community of online educators as to whether online assignments ought to be held to the same standards of writing that formal "on land" academic work is. It is my contention that doing so serves the students' interests best, helping those who are weaker in writing skills develop with frequent practice, and honing the skills of the proficient.

The four pillars of learning are *gathering data*, *reflection*, *creating* (using the learned concepts), and *testing* (testing one's new understanding in a social setting including other students or with an instructor).[4] A strategy I have employed in online classes is to have two discussion threads each week that students are required to participate in.

The first discussion thread focuses on theory; checking the student comprehension of the assigned text, research topic, or instructor commentary material. My goal is to guide all students to a certain baseline of comprehension of course material so that they are ready to apply, reflect on, and evaluate it.

This is accomplished in the first discussion line by asking questions to reveal how well the student understood the material. I respond quickly to each student as he or she answers the questions, guiding further thought and exploration. The result of this one-to-one tutoring is that students at all levels get some benefit from individualized attention. Those who are at a more complex level of learning are pushed to "take things up a notch." Students who are struggling with basic comprehension are guided into increasing, and becoming comfortable with, their understanding of the material.

In the second discussion line, I ask students to work together in pairs or groups to apply the newly learned material to everyday problems. In this way, they can integrate what they have learned with their existing beliefs and see "what happens next." Students quickly become comfortable with these tasks and seem to enjoy them a little more than the first discussion line, maybe also because the instructor presence is not so obvious. I tend to stay out of sight unless discussion wanders off track or there is an outbreak of incivility.

The cycle typically ends with a short quiz, and the assignment of the next week's reading material. What I have found is that students are often exhilarated by the interactions and by the development of their skills and ideas. Also, the quality of the discussion is much more reflective than I typically see in a face-to-face class. The online venue forces students to formalize their thoughts as they write them out, and allows them time to reflect on their answers before they give them. The interpersonal discussion may also reach greater depth as students are allowed time to consider the ideas of their peers before responding.

Hybrid Course Delivery Strategies

This year I have also been experimenting with hybrid courses to increase student outcomes. It seemed to me that students would benefit from the more immediate feedback on assignments and the one-on-one student-instructor interaction that the online venue offers. In past face-to-face courses it has been a struggle to find ways to have students come to class well prepared, having read the assigned material, with a good understanding of what they have read.

To this end, I would hand out worksheets for students to complete on the targeted material to turn in at the beginning of class. Unfortunately, this was not an optimal solution. Sometimes the worksheets were not completed at all, or when I reviewed them I found significant errors in understanding course material; errors that tended to persist, even after the lectures and class activity. The lectures themselves seemed at times to be less interactive and more "catch up." Part of the problem seemed to be a matter of catching the errors before they were firmly embedded in memory.

With a hybrid course, I tried a delivery model that has the same one on one student/instructor interaction for establishing a baseline of comprehension through guided discussion threads. I found that students often bring questions that arose in the discussion thread into the classroom discussion and the face-to-face component has become less of a lecture and more of a discussion. With the preparation work done online, less class time is devoted to basic concepts and more can be spent in application and evaluation of ideas.

At the end of the week I have a short quiz which integrates material from the texts, the online discussion, and the face-to-face discussion. This encourages students to attend to all delivery venues rather than just the online, or just the face-to-face. I am satisfied with these hybrid course experiments, and I think that a blended venue has some definite benefits for student learning and developing learning strategies.

One Final Observation

There is one little benefit of online and hybrid course work that I have not mentioned so far. That is that while there is an initial "front loaded" amount of work in developing a new course, once that is done, it is very easy for an instructor to keep up with grading assignments, to monitor student progress, and to keep his or her finger on the pulse of the class. This allows for more reflective interaction with the students. Online learning can be a gateway for reflective student learning, yes, but also can allow more challenging and interesting interaction for instructors as well. And in the end, that may be why many of us started teaching in the first place.

Endnotes

1. Bloom, *Taxonomy of Educational Objectives*.
2. Cozolino and Sprokay, "Neuroscience and Adult Learning," 11.
3. Baker, "The Impact of Instructor Immediacy and Presence for Online Student Affective Learning, Cognition, and Motivation."
4. Zull, "Key Aspects of How the Brain Learns," 3-9.

Bibliography

Baker, Credence. "The Impact of Instructor Immediacy and Presence for Online Student Affective Learning, Cognition, and Motivation." *Journal of Educators Online* 7, no. 1 (2010).

Bloom, Benjamin. *Taxonomy of Educational Objectives: Handbook I: Cognitive Domain.* New York: David McKay, 1956.

Cozolino, Louis and Sprokay, Susan. "Neuroscience and Adult Learning." *New Directions for Adult & Continuing Education* 110 (2006): 11-19.

Zull, James E. "Key Aspects of How the Brain Learns." *New Directions for Adult & Continuing Education* 110 (2006): 3-9.

Distance Learning and Philosophy: The Term-Length Challenge

Thomas Urban
Houston Community College

Higher education today, for better or worse, is all about securing the means to a more financially rewarding job, both among students and the institutions offering those educations. This attitude, coupled with a rapidly changing global economy and a public perception that "good jobs" are few and far between, has resulted in students feeling increased pressure to obtain that education quickly and with the least amount of difficulty and disruption. Getting on with getting a job and the pursuit of career goals is seen to be correlative with jumping on a fast-track to completing degrees and certificate programs. The question for educators is how best to address that pressure while continuing to maintain sound standards of student success and professional integrity, which together provide the basis for a sound learning environment.

The following discussion and report focuses on one way this question is being answered with the help of online technology. Specifically, it considers how shorter and shorter time-frames for college courses may or may not be benefitted according to their proposed length, the scope and parameters of the technology employed, and the subject area being taught. Of particular interest are those courses offered through humanities and social science programs and how the demand for a compressed education pressures institutions to find more efficient ways to deliver their products in ways that match the expectations of all constituencies. Those constituencies include students, teachers, administrators, politicians, and the general public.

Section I considers several key questions that I raise and respond to with an aim to flush out the real possibilities for the most effective uses of online technology as it combines with differing course content and lengths. In thinking this through, I've found it necessary to bracket and set aside those criticisms of online education that proceed from an assumption it's inherently inferior. Inferior examples of course delivery can be found in every teaching and learning format. At the same time, I've noted that in some instances, the misuse or overuse of online technology can be detrimental to learning, the danger when thinking loses sight of technology's virtual nature. Let's not be naive. There are ways to think about and use technology effectively, and ways not to do so. The responsible aim of educators is to discern that difference. Section II focuses on the case of a "pilot" three-week distance learning section of Introduction to Philosophy that my colleagues and I have been monitoring over the past two years. As a pilot course, no particular assumptions accompanied its initial development beyond a perceived demand for an online course of this decidedly short length. At this juncture, student learning outcomes for this course have made it clear that the concept does not quite align with best practices in philosophy when compared to longer terms and competing online formats, e.g., web-enhanced and hybrid for courses of the same length. Our judgment is not to continue authorizing this term-length for that particular course. At the same time, our focus is now directed to developing new guidelines for offering distance learning and other web-assisted courses in lengths that vary greatly from past conventions in terms of both delivery and curriculum. This task will require that we not only consider what it is we actually teach in our courses, but also how to align that content and teaching with those forces within and without higher education that challenge our beliefs about the right thing to do for students.

I.

The impetus to move to shorter terms for classes finds its rationale for students from three perspectives. One is the desire of students to complete their education in as short a time-frame as possible. This could be due to simple economics, e.g., taking classes cuts into the time one can devote to earning money or the sooner an educational program is completed the sooner one can advance to a better, more lucrative position. It could also be due to the fact the student doesn't really enjoy taking classes, even when those classes are associated with career goals. Third, a shorter time-frame for courses has been shown in research to produce more positive student learning outcomes, an indication that students are actually obtaining something positive from the experience. One reason that this is the case is that the traditional long 16-week semester constructed around a 15 to 18 SCH schedule begins to lose focus after 10 to 12 weeks. Though many colleagues would disagree with this assessment and argue for the elimination of all shorter term offerings, I have long noticed a perceptible drop in energy by the twelfth week, a drop that doesn't appear in the shorter terms. While my observations are strictly anecdotal, studies are readily available that illustrate the same conclusion, e.g., a comparison of course length to developmental student success at http://www.informaworld.com/smpp/content~db=all~content=a918017330 .

The main question that I would raise with respect to studies finding more positive learning outcomes relative to course length is to ask what kind of learning is being measured. If one is talking about information, even concepts that can be memorized by rote, the learning while measurable and useful is not the same as the kind of learning more commonly associated with philosophy and more general categories of the humanities and social sciences. Learning in these areas invariably entails both arts and skills and an evolving *ethos* for social and political understanding. In other words, rote learning can take place with neither a clear sense of how it came to be in the first place, nor how it applies to life experience. It requires little or no teacher-student engagement to be acquired by students deemed college ready. Courses in areas that have such learning as their primary emphasis include many that are basic to science and workforce or technical programs, and that typically exhibit this character.

On the other hand, courses typical of the Humanities and Social Sciences that have the cultivation of an open, critical mind as their primary educational goal all require teacher-student engagement. Students in these areas need someone to point out alternative ways of thinking and challenge those unnoticed habits of thought that blind us all from seeing the possibilities of a richer, more fruitful understanding for thinking and the constitution of our social consciousness. Though the tools for accomplishing this activity may differ between a face-to-face class and one conducted online, there is no difference in substance when the available tools are in fact being used. A common error is to confuse obvious differences in delivery with other contributing factors that can be found in any learning context, e.g., laziness, lack of self-discipline, and lack of interest. Given this, it is reasonable to ask whether a very short time-frame inhibits a teacher's ability to identify and address these other factors or make optimal use of the tools she or he has in hand. In other words, does an instructor in a face-to-face class have an advantage over one teaching either a hybrid section or a course that is entirely online? How long does it take a traditional face-to-face instructor to identify a student having difficulties? Can online technology be set up in ways that help make the timeliness of that determination co-equal or better than that traditional approach? Put another way, can online technology be employed in ways for distance learning and hybrid courses to offset any perceived loss of direct face-to-face contact when 16-week courses are reduced to as few as three-weeks, and everywhere in-between? Experience with and outcomes for eight to twelve-week courses suggest it can, even when that use is merely auxiliary as is the case with web-enhanced classes. And, web-enhanced three-week courses show themselves to be more effective than three-week courses with no web-presence whatsoever. If nothing else, student tracking is ongoing and tends to be more precise, which opens the possibility of better oversight of student activities.

Though unrelated, another important impetus to scheduling classes in a shorter time-frame is to generate revenue for institutions that are constantly challenged to maintain an adequate revenue stream to meet the needs of their educational mission. The argument here follows an assumption that shorter courses allow for a greater number of offerings over a year, and shorter courses are more attractive to students, which will increase enrollments, and with an enrollment increase one finds a correlative revenue increase. Current trends to cut subsidies to public colleges and universities, reduced charitable giving to private non-profit institutions, and reluctance to simply raise tuition to make up shortfalls are three compelling reasons to restructure programs in ways that accommodate shorter courses. The most successful of these shorter courses are for the most part using some degree of online technology. That is, they include courses in all three online delivery formats but not necessarily all three in combination with each of the various lengths of terms one finds. For example, an eight-week course may work well as web-enhanced, hybrid, or distance learning, whereas a five-week course may be unsuited to all except web-enhanced, depending on the subject area. The National Center for Academic Transformation (NCAT) has an online resource page at http://www.thencat.org/ that provides both survey results by subject area in these regards as well as information on how to alter course design in ways that incorporate the effective use of technology relative to course length and student success.

Traditional face-to-face courses have been offered in a wide variety of time frames for years, e.g., the long semester, the "quarters" system and its variations, and the three-week mini-session. Still, in my experience as both student and professor, I have yet to hear anyone question the integrity of any of the shorter versions. The only thing I can note is that with shorter courses there has always been a correlative lowering of the maximum number of credit hours a student may register for within the shortened term. This absence of controversy has not been the case with distance-learning courses. Indeed, there has been great controversy about the practice of offering either hybrid or distance learning courses in these shortened time-frames, particularly those that are set for fewer than 10 weeks. One can only wonder about the reasons for this opposition, except to suggest the course architecture and delivery appears so different to the untrained eye that the eye is unable to see how these courses could possibly be as effective as traditional face-to-face offerings of equal length.

More important is the thought that when creating the architecture for any course of any length, face-to-face, web-enhanced, hybrid, or distance-learning, one must create a student-centered environment. This means that the learning environment, whether real or virtual, must address the essential need for the continuous engagement of thinking by both the student and teacher. And, while some kinds of learning may be successfully accomplished without doing this, it is not the kind of learning called for by either the humanities or social sciences. How to do this is a challenge, not to mention a great deal of work in creating the course and then in carrying out its delivery. Where technology is introduced, one must imagine the virtual spacing of the course in such a way that both its content and called-for activities provide the necessary conditions for the real possibility of every student's success in that course. Along with imagining the virtual spacing of such a course is the need to seek out a learning management system that will allow its actual construction. Not all learning management systems do this, or if they do, they require a great deal of "outside-the-box" thinking to make them match up with both student and teacher expectations.

II.

The following account concerns a pilot three-week distance learning section of Introduction to Philosophy offered by Houston Community College, including a description of an adopted plan for altering course architecture with the aim of making it more student-centered. The learning management system used by the instructor of this pilot course is Blackboard Vista and the course itself represents an updated version of a course originally written for WebCT. As earlier stated, the evidence relative to our discipline's student learning outcomes (SLOs) suggests it does not compare favorably with on-campus face-to-face sections of the same curriculum and length or distance-learning sections that run for eight or 10 weeks. The pilot test concluded at the end of the Spring three-week term in early June 2011. A committee of full-time colleagues evaluated the data and are responsible for the decision not to continue the course. Interestingly, the administration, which has pressured disciplines in the past to offer distance-learning courses during the three-week session, also noted a significant 20 percent drop in student success rates for distance-learning sections when compared to their face-to-face and web-enhanced three-week counterparts. This was an observation made across the curriculum, not just in relation to the three-week philosophy course. What's even more interesting is why they were comparing these outcomes. They were considering a cost-saving measure that called for canceling the on-campus courses, but when they crunched the numbers found that the on-campus sections were in fact more profitable due to high rates of student withdrawals as well as failures in the distance-learning sections.

Houston Community College offers an average of 80 sections of lower division philosophy courses each Fall, Spring, and Summer. Among these are 17 distance-learning sections of four courses, *Introduction to Philosophy*, *Ethics*, *Survey of Ancient and Medieval Philosophy*, and *Women in Philosophy*. A new distance-learning course, Political and Social Philosophy, is being introduced in Fall 2011. Until Fall 2008, no distance-learning section of philosophy was offered in less than the 10-week Summer term. The Summer term was considered to be the shortest time-frame in which it was possible to assure student success in distance-learning courses. Concurrently, HCC regularly offered five-week face-to-face Summer courses as well as three-week face-to-face "mini-term" sections going back several years, one in December, the other in May. Then, our Discipline Committee discovered that two face-to-face sections of *Introduction to Philosophy* had been rescheduled without our knowledge and ran as distance learning courses in December 2008. This same thing happened again when two 10-week and one eight-week Summer distance-learning sections were switched to five-week courses, again without discipline consultation or approval. Why this occurred without our knowledge, the issues it raised, ensuing controversy, and where the matter stands now is the subject of this report.

Administratively, philosophy does not have its own departments or department chairs at HCC's six autonomous colleges. Across the HCC system, our courses are offered through five of those six. Oversight of these courses is shared by a system-wide Discipline Committee of full-time philosophy faculty and five department chairs, none of whom are philosophers. The department chairs schedule our courses. While our discipline chair works closely with them, the chairs often find themselves under pressure from administrators to offer courses in formats that do not necessarily agree with discipline standards or previously established college precedents. This has been particularly true with distance learning, where the demand for shorter courses has prompted many disciplines to now offer sections of not only three-week mini-terms, but also five- and eight-week terms and "Ready When You Are" courses. The mentioned 8-week distance-learning course offered in philosophy during two previous Summer sessions was another example of a course that had originally been scheduled for 10 weeks. This same drift toward offering shorter courses is also evident in face-to-face scheduling with the announcement in January 2011 that a select number of core courses will now be offered in eight-week sessions with the expectation that all disciplines adjust their schedules to include these shorter offerings in all delivery formats. If this initiative succeeds without any documented deficiencies in student learning outcomes, there is no question that the administration will argue for scrapping longer terms as a way to serve a greater number of students over the calendar year, a move that would also significantly raise revenue. While an eight-week time-frame is not comparable to offering the same course over three-weeks *per se*, there is a real challenge in devising effective ways to deliver the same content that is currently delivered over 10 to 16 weeks if student learning is our primary objective. Accreditation guidelines prescribe those courses carrying the same title and credit must be of like content, regardless of length and delivery method. In other words, some of the same issues that confronted the development of distance-learning courses alongside on-campus offerings are now issues for faculty teaching face-to-face courses, including the requirement that an effective use of technology is essential to address those issues.

While some may question the wisdom and reasons underlying this demand for shorter course formats and greater flexibility, the fact is that all of these courses tend to fill up early during registration, particularly the distance-learning sections. For example, when the five-week distance-learning sections of Introduction to Philosophy appeared on the Summer 2011 schedule, they filled weeks before the competing 10-week sections. From the perspective of administrators and some department chairs, short to medium distance-learning courses are a win-win on all counts. The one question that remains is whether it is a win for students and for philosophy, which is where discipline oversight can prove helpful. While students who are of a mind to attain their program objectives as soon as possible, it is the discipline that is charged with assuring that achieving established benchmarks in our courses is part of that attainment.

The HCC Philosophy Discipline has always promoted the use of technology in the delivery of our courses. Across the system, most philosophy courses are web-enhanced with a Blackboard learning management system (LMS) presence, supplemented by customized "Learning Web" pages maintained by faculty. Many of those sections also include a component for online course credit, though less than hybrid courses. We actively encourage our entire faculty to complete HCC's very rigorous "Certificate in Teaching Technology" program, which permits them to teach distance-learning sections. In addition, no new version of any of our distance-learning courses can go online without a thorough review by the discipline/program chair, who through 2010-2011 was an individual who holds the

HCC "Advanced Certificate with Portfolio Review" from the teaching technology program. Our incoming chair is a newly trained distance-learning instructor. Finally, three of the four developers of a humanities distance-learning "Master Course" were members of our Discipline Committee. In other words, our concern is not a question of rejecting new uses of technology on principle.

However, we do not support the careless or haphazard use of technology. This is what we feared was the case when we discovered that one of our colleagues had opted to teach the two December 2008 mini-term sections of *Introduction* in a distance-learning format. The resulting strain it placed on otherwise collegial relations among committee members continues to haunt our interaction, mainly because any *ethos* of trust was broken. It was also not helpful that half of the committee was vehemently opposed to the practice on grounds that our curriculum is too extensive to deliver online in such a short time-frame with severely limited instructor-student contact.

This opposition may indicate a bias of sorts against distance-learning courses. But again, the established HCC standard is that distance-learning courses are to be every bit as comprehensive and rigorous as our face-to-face sections. In particular, that rigor assumes continuous direct engagement and feedback by instructors to individual students. Those on our committee who were opposed to the three-week format were of the opinion that an instructor teaching such a course would of necessity find him or herself online 24/7 to achieve the same results as an intensive three-week face-to-face class. The instructor who was teaching those sections saw no problem since that person simply squeezed the course calendar used for the 16-week semester into three, and explained that serious students seem to have no difficulty handling the material, making no mention of the instructor's role. Assuming that there was no difference in the way this instructor graded and commented on the work of mini-session students, that claim is at least partly supported by the final grade distributions. Twenty percent received "A," 22 percent "B," and 13 percent "C," while 45 percent of the students originally enrolled in both sections either failed or withdrew. Similar percentages were reported for subsequent three-week terms for these pilot courses, with one term showing over 50 percent of the students enrolled either did not complete, or failed.

But, when these distance-learning numbers were compared to face-to-face three-week sections of the same course, a very different result showed itself. Face-to-face students who either withdrew or earned less than a "C" ranged between 10 percent and 13 percent. Furthermore, that range is comparable to rates found in our distance-learning courses offered over longer periods. However, grade distributions and retention rates alone do not give a complete picture of what's going on here. Our question was and continues to be whether there is a better approach by which to design a three-week distance learning course, the aim being to improve the numbers. While students must bear responsibility for failing to successfully complete courses in which they enroll, there is a correlative responsibility for instructors and institutions not to set them up for failure or withdrawal by promising fast learning opportunities without simultaneously underwriting those opportunities with effective learning strategies. The counter argument advanced by those who see nothing wrong with the distance-learning outcomes is that it's the students' responsibility to successfully complete the courses as offered, an argument our discipline committee rejects.

At the time this issue first appeared, our committee voted to prohibit offering distance-learning courses over the three-week session. However, the instructor's dean and the vice chancellor of instruction countered that decision and allowed the practice to continue. We held this to be a clear violation of established rules that mandate a discipline committee's authority to oversee the delivery of courses, rules that conform to Southern Association accreditation guidelines. But, rather than lodge a formal complaint with the Southern Association (SACS), our committee instead decided to restrict the practice to that one instructor's courses, and to turn the delivery of those courses into the pilot project that ended in June 2011. It was clear that forces beyond our control and the control of administrators had gathered in a way that is directing all to rethink course content and delivery. It was also clear that technology is not the culprit many believe it to be. The culprit is thinking that has yet to master the impact of those forces and ways that technology may prove truly useful to student learning.

Accordingly, the aim of our pilot project was to continually review and alter the distance-learning course architecture for the purpose of creating a viable three-week course, namely, one that at least approximates student learning in our face-to-face three-week sections. While the result merely confirmed the committee's worst fears, we do not believe the approach was flawed; only that a three-week time-frame is too short to preserve a viable learning environment for our established curriculum. The pressure to develop both distance-learning and on-campus courses that provide students a sound learning environment in shorter terms has not abated and is a reality that calls for continual review and alteration. At bottom, the question is one of delivery. Integral to this for us is our discipline's implementation of a course-based outcomes assessment (OA) plan. It is one that makes both learning outcomes and objectives for each of our courses explicit in a way we believe can be utilized by teaching faculty to alter their delivery of our curriculum. This alteration is achieved by a process called "backwards design" (see Wiggins and McTighe, *Understanding by Design*). Learning outcomes under our assessment plan are measured by Standard Form rubrics that are then reported for a system-wide assessment at the end of Fall, Spring, and Summer, which also provide the basis for student-generated program learning outcomes (PSLOs). Every course in our 20-course inventory has five course-specific Student Learning Outcomes (SLOs). The actual assessment of student learning according to our Standard Form Rubrics is by the faculty member teaching those students. Learning outcomes are not necessarily correlative with grades, and this view is being stressed as we educate our faculty on how to conduct their assessments. One thing we are already learning is that this means of assessment is pointing out real differences in how a distance-learning instructor is challenged to arrive at a candid assessment, including the kinds of assignments she or he uses. As one distance-learning instructor pointed out at the end of Spring 2010, a major essay is not necessarily an indication of student learning since the distance-learning instructor has no way of knowing for certain who wrote the essay. Other kinds of assignments may prove more effective in evaluating the true abilities of the students enrolled and thus provide a better basis for honest measures, namely, assignments that require ongoing faculty-student interaction.

Since the implementation of this other means of assessing student learning is still in its early stages, the jury remains out as to its overall impact and whether it will prove useful in developing other than three-week distance-learning courses that meet discipline standards and expectations. After gathering four semesters of data, what is most obvious is the learning-curve required for faculty to understand how it works since it is a separate measure from earned grades and retention. More important to the subject of the question at hand is how it can work to create more student-centered courses, no matter what the length or delivery method. We believe our plan shows promise in these regards and will ultimately provide a fair-minded judgment on the issues involved. It is certainly more robust than a judgment based solely on grades and retention rates, and is a promising means toward overcoming the bias of those who view change as "dumbing-down" curriculum. Without having an ongoing comprehensive course-based outcomes assessment plan in place, we would have little of substance to consider. But, more is needed. While our OA plan provides instructors with SLO-specific objectives that can be used to effectively alter the architecture and delivery of their courses, training in the principles of backwards design is a necessary next step. A related concern is to show faculty that this plan as developed is not an infringement on their academic freedom and does not require them to sacrifice either their standards or passion. Following our judgment that the pilot three-week distance-learning course was a failure as conceived, the reasons for that failure are not entirely clear despite the outcomes on all fronts. It is altogether possible that new guidelines for curriculum and course development (and delivery methods) that better incorporate the expected outcomes of courses can and will result in finding ways to adapt those same courses to any number of term lengths. What is clear is that the thinking required to monitor that pilot along with the development of a comprehensive yet flexible Outcomes Assessment plan places our philosophy program in a far better position to consider restructuring it along the lines of shorter courses for both face-to-face and online distance learning. A potentially helpful development is HCC's move to a new Learning Management System based on "Moodle Rooms" that appears promising. Should eight-week sessions become the norm at our institution, including eight-week hybrid courses, it is also imperative that a continuing effort be made to maintain our OA plan's effectiveness. The integrity of our program and future for teaching philosophy in a rapidly changing learning environment demand it.

Conclusion

The only conclusion that might be drawn from the preceding discussion is that the jury is still out as far as determining the precise combination of course length and online technology for its optimum application to courses in philosophy, and more generally to courses in all of the humanities and social sciences. What is presented is an idea for how to think about making such determinations by combining best practices in teaching and learning with course design and the selection and use of effective tools for ongoing student-teacher engagement. Student expectations that courses be offered in shorter and shorter time-frames must be moderated with the expectations of teaching faculty. Faculty expectations must be embodied in measurable outcomes showing real student success in the subject area and a willingness to adopt correlative methods of delivery. What I see emerging is an ongoing transition and development towards a very different kind of college and university experience, one that will continually transgress conventional modes of course content and teaching strategies until such time that a new sense of equilibrium is established. Meanwhile, the presenting challenge is to seek out the best available tools from all sides, to meet the challenge, and build for the future.

The Heritage of Gaetano Aurelio Lanzarone

Federico Gobbo
University of Insubria, Italy

The night of October 3, Gaetano Aurelio Lanzarone, full professor in Computer Science at the University of Insubria, Varese, Italy, passed away. He suffered from an incurable illness since at least 2008, but he never missed a day's work, except in the very last days. I had the opportunity to work closely with him during the last years (2002-2010) of his very long academic and professional career. What I want to do here is to give a memoir of his career, with special attention to my personal experience with him, while his smile, laughter, and humor cannot be described in words, but only in the language of our hearts and memories.

Born in Palermo (Italy) in 1945, Elio—as close colleagues and friends called him—obtained his Master's degree in physics in 1970 at the University of Milan, Italy. At that time, there was no educational curriculum in Computer Science in Milan; even if he once said that "Physics is the Queen of Sciences," his Master's thesis was in Computer Science, his real field of interest. In fact, he started to work as a consultant for R&D Labs of Honeywell Information System Italia until 1976 and then he became the head of the Software Methodologies Group at the R&D Labs of Italtel from 1976 to 1985—a very important Italian Telecommunications Company in those years. His first academic experience was lecturing in the courses in Cybernetics and in Computer Science at the University of Milano, during the academic years from 1974 to 1978, and from 1978 to 1985 as a tenured professor. From 1982 to 1984 he was a member of the group of European experts appointed for the definition of the European research programmes JEPE-IT/ESPRIT (Joint European Planning Exercise - Information Technology / European Strategic Programme for Research and Development in Information Technology) and PET-RACE (Planning Exercise in Telecommunications - Research and Development in Advanced Communications Technologies in Europe). For at least a generation of Italian computer scientists, Lanzarone is still known nowadays as one of the first people who introduced structured programming in Italy, after the famous Dijkstra's dictum in 1968; in fact, he co-authored a very influential book about this topic (Maiocchi, Lanzarone, and Polillo 1986).

In my experience, Lanzarone was always curious to explore new fields of knowledge and to have new experiences and opportunities to grow. I remember an informal talk with him about this attitude of mind that causes his turn of interest from software engineering to Artificial Intelligence in the 1980s, a field in which he is considered a pioneer in Italy. When he became full-time associate professor at the University of Milan, in December 1985, he quickly founded the Laboratory of Logic Programming. His major research scientific result—in my opinion the most important in his whole life, obtained with Stefania Costantini (University of L'Aquila, Italy)—is Reflective Prolog, which is an augmentation of pure Prolog (Horn clauses) with capabilities of self-reference and logical reflection, using quotation and unquotation mechanisms (Costantini and Lanzarone 1994). In the 1990s he applied his scientific results in the field of knowledge engineering: in particular, he used logic programming in order to represent mind models belonging to Cognitive Science computationally, in particular with Stefania Bandini (University of Milano-Bicocca). For instance, he had co-authored a paper where the theory of induction by Johnson-Laird was modeled in terms of logic programming (Bandini, Lanzarone, and Valpiani 1998).

However, in spite of his affiliation to University as a professor, he never lost attention to the collaboration between University and ICT industry, being scientific coordinator of several research projects funded by different public and private institutions, at any level, from EU programmes to Lombardy. In particular, he was responsible from 1995 to 1998 for the participation to the European Consortium of the project THAIland (Telematics, Hypermedia and Artificial Intelligence), funded by the European SOCRATES program, for the definition of a Master Curriculum in Telemedia Studies.

Another important trait of the personality of Lanzarone was his firm belief that the dichotomy “humanities vs. sciences” is false: humanists can profit from the results of sciences, and vice versa, he used to say, without unnecessary hierarchies between the two groups. At the University of Insubria, his idea of fruitful cooperation between humanists and scientists could became reality: for example, he published some papers with an art historian (Benini, Lanzarone, and Spiriti 2007). At the end of the 1990s, the newborn University of Insubria (Varese-Como, Italy) gave him the opportunity to establish first (2000) the Research Center “Informatica interattiva”—meaning both “informatics in interaction” and “interactive informatics”—and then (2004) the Department of Computer Science and Communication (DICOM), where he was Head until the dismission of the Department at the end of September 2011.

From March 2000, as a full-time full professor, he started the undergraduate curricula in Computer Science and Communication Sciences. I met him first in 2002 as a R&D consultant of a start-up company, Pumpkin Srl, for a couple of projects where DICOM was partner about Digital Cities and a virtual gallery of an exhibition of the portrait in Lombardy in the cinquecento. I already had a Master in Communication Sciences (1998) and another in Computer Science for the Humanities (2000): soon after our first meeting, I entered his team, working in particular with Marco Benini—now a Marie Curie Fellow in mathematics, University of Leeds. After 2003, I had a couple of research grants at the crossroad of Computer Science and Linguistics, and completed my education with a Ph.D. in Computer Science. In both cases the supervisor was him. But I collaborated with him not only for research but also in lecturing. In fact, he decided to open a course for graduate students about Computing and Philosophy, whereas a correspondent course for undergraduates about History of Computing was established, for students in Computer Science and Communication as well. I actively collaborated with these two groups of students as a tutor alongside Lanzarone, while being lecturer in the last two academic years, after finishing my Ph.D. in January 2009. During this experience, we made an experiment of collaborative learning and philosophical writing among students using a wiki (Gobbo and Lanzarone 2006). His perspective on the discipline was published rightly in this *Newsletter* (Lanzarone 2007). His research interests seemed never to stop: in fact, thanks to his support, I was able to co-found the European Summer School in Agile Programming (ESSAP), a particular approach to software engineering, which “resembles the old good common sense present in the early days of software engineering when I was involved in structured programming,” he once told me. ESSAP lasted three editions, calling Ph.D. students, graduates, and professionals from every part of the continent to the beautiful Villa Toepliz in Varese.

However, in the last years his main interests were epistemology of computing and computer ethics as well. From 2005 to 2008 Professor Lanzarone was active into the community of researchers in Computing and Philosophy: in spite of the short years of engagement, he not only participated in various conferences (E-CAP and NA-CAP in 2007, ETHICOMP in 2008, where we jointly presented a contribution, see Lanzarone and Gobbo 2008) but also was co-organizer and invited speaker—alongside Peter Boltuc, Keith Miller, and Vincent Müller—to the first Computing and Philosophy Global Course, a result of collaboration between several European and American universities, based on an earlier Swedish National Course organized by Gordana Dodig-Crnkovic.

Lanzarone’s heritage is great in people, like me, who had the chance to work closely with him. Farewell, Elio. Have a nice last trip.

Essential bibliography of Lanzarone’s publications

Bandini, Stefania and Lanzarone, Gaetano A. and Valpiani Alessandra. 1999. “Revisiting the Mental Models Theory in Terms of Computational Models Based on Constructive Induction.” Special issue “Analogy and Mental Modeling in Scientific Discovery,” *Philosophica* 62, no. 2 (1998): 901-23.

Baiocchi, Marco and Lanzarone, Gaetano A. and Polillo, Roberto. *Introduzione alla programmazione strutturata. Il caso del Fortran, Cobol e Assembler*. Milan: Franco Angeli, 1986.

Benini, Marco and Lanzarone, Gaetano A. and Spiriti, Andrea. “Navigazione tematica di un museo virtuale: una soluzione mirata.” *Conferenza GARR 2007 Network Humanitatis - un filo di luce che unisce il sapere*, Rome, Italy, October 29-31, 2007.

Costantini, Stefania and Lanzarone, Gaetano A. “A Metalogic Programming Approach: Language, Semantics and Applications.” *J. Expt. Theor. Artif. Intell*. 6 (1994): 239-87.

Furlan, Franco and Lanzarone, Gaetano A. *Prolog. Linguaggio e metodologia di programmazione logica*. Milan: Franco Angeli, 1988.

Gobbo, Federico and Lanzarone, Gaetano A. “A Wiki-based Active Learning System; How to Enhance Learning Material in Epistemology of Computer Science and Computer Ethics.” *Proceedings of Current Developments in Technology-Assisted Education*, Volume II. Badajoz (Spain): FORMATEX. (2006): 757-61.

Lanzarone, Gaetano A. and Gobbo, Federico. “Is Computer Ethics Computable?” In *Living, Working and Learning Beyond Technology. Proceedings of ETHICOMP 2008*, edited by Terrell Ward Bynum, Maria Carla Calzarossa, Ivo De Lotto, and Simon Rogerson. Mantua: University of Pavia, 2008, 530-539.

Lanzarone, Gaetano A. 2007. “Computing and Philosophy: In Search of a New Agenda.” *APA Newsletter on Philosophy and Computing* 07, no. 1 (2007): 18-21.