Untitled Number 4: A Brechto-Socratic Dialogue

Stephen Ramsay and Geoffrey Rockwell

March 18, 2004

ROCKWELL: After this session let us leave and walk along the banks of
the Ilissus; then we can sit down in any quiet spot you choose....
It's convenient, isn't it, that I am barefoot; you of course always
wear sandals, so we can wade in the stream, which is especially
delightful on a hot summer Georgia day (Plato, 1989).

RAMSAY: But alas, there's no Ilissus here—just the Oconee, which isn't
really the same thing. You must have the wrong Athens. And
anyway, I'm wearing shoes!

ROCKWELL: I'm paraphrasing the *Phaedrus*, Ramsay. Don't you re-
alize that this is the twenty-five-hundredth anniversary of the
Phaedrus?

RAMSAY: It is?

ROCKWELL: Well, not exactly. But, as several people have remarked,

it's not really the tenth anniversary of the World Wide Web either and that isn't stopping us. And we are in *an* Athens, even if we are some 4,875 nautical miles from the real one. The *Phaedrus*, at any rate, is entirely relevant to our paper. As they say, all humanities computing is but footnotes to the *Phaedrus*.

RAMSAY: Wait, hold on a minute. This isn't going to work. Stop the recording. This is how we began the dialogue when we delivered it at the ACH conference here last summer. No one is going to get the joke about Athens and the 10th anniversary of the Web when we put it online. They're not going to know that the Oconee is our local river, or that the name of the conference was "Web X: A Decade of the World Wide Web?" I think we have to rewrite the beginning of the script and start over. We've got to make corrections anyway.

ROCKWELL: I thought what we were doing was digitally recording the dialogue as originally scripted to make it available? This script is the original, lets not introduce changes between the PDF of the script and the audio performance.

RAMSAY: But who cares about the original? It's not like it was rare manuscript, where we need to encode all the changes over time.

ROCKWELL: This project is getting out of control. Patrick, did you stop recording?

KOVACICH: No, I didn't hit the pause button. We have lots of tape.

ROCKWELL: Well, let's just keep on going. We'll put some explanation or something on the site to provide the context. Where were we? Ramsay, you were going to introduce the theme.

RAMSAY: Our theme is love.

ROCKWELL: No, no. You're thinking of the first part of their session, you know, that set of three speeches. The second part concerns writing and rhetoric and is a dialogue reflecting back on the speeches. They end by agreeing how speech-writers like Lysias should be approached and reproached about writing. As Socrates says, quote, "And now to revert to our other question, whether the delivery and composition of speeches is honorable or base, and in what circumstance they may properly become a matter of reproach...." (Plato, 1989) In the end, they conclude that speech-writers like Lysias should be approached and reproached through questions about writing.

RAMSAY: That hardly seems pertinent to the theme of the conference.

ROCKWELL: Oh, I think it is. The *Phaedrus* inaugurates a centuries-old discourse around the technology of writing and writing as technology which leads to our topic today.

RAMSAY: Which is?

ROCKWELL: Writing as programming as writing. Remember the title we sent in so many months ago?

RAMSAY: Well, I can certainly see the relevance of that bit where

Socrates tells Phaedrus the story of the invention of writing and its judgment. It's a grand tale. And if I recall, Socrates ultimately reminds us that writing, as cool a technology as there ever was, is not a recipe for wisdom. It's a story that's been retold dozens of times by critics of technology like Neil Postman. Books like *Technopoly* and Birkerts's *The Gutenberg Elegies*, after all, are really out to judge again whether computers or, for that matter, the Web have made us any wiser. So yes. Ten years of the Web. Time to return from Troy—perhaps with Casandra in tow—and ask ourselves where we've been and where we might yet be going. But I don't see the relevance of the larger discussion in the *Phaedrus* of rhetoric and speech writing to computing and the humanities. What does Plato have to offer a computing humanist who wants to understand programming as writing?

ROCKWELL: Let me ask you a question. You love to write and you love to program?

RAMSAY: Yes.

ROCKWELL: When you write you produce text, and when you program you write code, right?

RAMSAY: Yes, Socrates.

ROCKWELL: Very funny. And tomorrow we are both on a panel on peer review of code as if it were text. Right?

RAMSAY: It is so.

ROCKWELL: Then we have a question which is suitable for this audience—a question which concerns code and whether code is in fact a form of textuality for which the rich tradition of the humanities around reading, dialectic, rhetoric, and reasoning are appropriate arts. Surely that is relevant to computing in the humanities—a discipline around the intersection of code and text—and, if we can determine the nature of code and text, we might then return to the issue of programming as writing. We could even agree, like Socrates and Phaedrus, that tomorrow we are going to reproach programmers who, like speech-writers, have forgotten wisdom for codes.

RAMSAY: Well, code and textuality is a favorite of mine. I've been hacking code all night (out there beyond the walls of Athens in the county) and I'm quite convinced that code is a form of textuality that operates according to the terms you suggest.

ROCKWELL: That's a pity, really, because I'm going to argue precisely the opposite.

RAMSAY: But I don't think you actually believe the opposite, Rockwell.

ROCKWELL: So? This is the game of philosophy.

RAMSAY: Well, who goes first?

ROCKWELL: Me, of course. I'm going to be Socrates.

RAMSAY: And I am going to be Phaedrus, I suppose. I seem to recall that neither of them got published in peer-reviewed journals, and

Socrates was poisoned for corrupting the youth of Athens with questions.

ROCKWELL: We seek not publication, my dear Steve, but truth.

RAMSAY: Obviously. Very well, then, why don't you start.

ROCKWELL: [*Straightening himself*] Well, let's start the way Socrates did, and I quote, "We must know what it is that we are deliberating about; otherwise, we are bound to go utterly astray. Now most people fail to realize that they don't know what this or that really is; consequently when they start discussing something, they dispense with any agreed definition, assuming that they know the thing; then later on they naturally find, to their cost, that they agree neither with each other nor with themselves" (Plato, 1989). That's from *Plato: Collected Dialogues*. Edith Hamilton and Huntington Cairns, eds. Princeton UP, 19—

RAMSAY: [*interrupting him*] I don't think you're supposed to give citations in an oral performance.

ROCKWELL: Well, whatever. I propose that we use a definition of code from a Web authority—dictionary.com—which reads: Code is "A system of symbols and rules used to represent instructions to a computer; a computer program" (*American Heritage Dictionary of the English Language*, 2003) Is that an acceptable definition?

RAMSAY: Yes, though a somewhat terse one.

ROCKWELL: Well we can develop it as we go. Now listen: Text is

also a system of symbols (and loose rules), but it is meant to be read not by humans, but by a computer. One of the differences between code and text is therefore the audience of the outcome of programming and writing.

RAMSAY: Yes, but code is also meant to be read by humans. That, in fact, is the primary reason why programming languages exist. From the computer's standpoint, the most legible language is electronic impulses—zeroes and ones, if you will. No one writes programs that way precisely because the code needs to be understandable not only by the computer, but by the person writing the code and by others who need to verify its logics and extend, modify, port, and replace them. One of programming's greatest teachers—Harold Abelson, a veritable Pythagoras among computer science professors—has even gone so far as to say that "programs must be written for people to read, and only incidentally for machines to execute" (Abelson, 1996).

ROCKWELL: But that executable function is surely far from merely incidental. In order to be executable, code has to be unambiguous in a way that we do not expect text to be. It has to be entirely unambiguous concerning, for example, what parts are comments and which parts are statements. The symbols, code words, operators and so on can have only one effect on the interpreter or the code fails. The code need not be legible by humans, but it has to

be legible to the computer if it is going to do anything useful at all.

RAMSAY: But human-illegible programs are likewise doomed to failure. It has become almost a truism to say that a program is a kind of narrative, in part because the accomplishment of a specified task is only one of several purposes for a set of coded instructions. We therefore describe coded instructions—rightly, I think—as the *story* of the machine's operation. We say that the program is the *specification* of that operation (as opposed to the operation itself). We speak of elegance, transparency, concision, readability—all terms properly associated with textuality.

ROCKWELL: Yes, but saying that a program is a *kind of* narrative or a *kind of* story doesn't make it so. You're just using those terms to be poetic, something you can do with text, but not code. Most programs aren't "stories" in any literal sense. If the average piece of code is a story or a narrative, then the directions for my microwave are also narrative. You surely don't want to say that ordinary instructions—for example, "go down to the Oconee and turn right when you see a water nymph"—are narratives?

RAMSAY: But don't you see that it *will* be a narrative when you follow those instructions. If I say, "Rockwell went down to the Ilissus, saw a water nymph, and turned right," I've surely related a narrative. Why not say that the directions—the code, if you

8

will—were the specification for the narrative?

ROCKWELL: Very well. But then, is the *Iliad* also the specification for a narrative?

RAMSAY: Yes. Absolutely. Let us say that the written text of the *Iliad* which the tyrant Peisistratus is said to have ordered made is not "the *Iliad*," but rather a set of instructions for the performances of the *Iliad* held at the Panathenaic festivals and then re-performed countless times in the centuries since.

ROCKWELL: Socrates had a word for this: sophistry. I don't "perform" the text of the *Iliad* when I read it—or, at least, not in the same way I perform a series of instructions. Code is explicitly designed to guide action, whether by a computer or a human. Let me try a different sense of code. Think of a code of ethics. It is a particular type of text designed to provide guidelines or instructions to the interpreter. If followed—where following is more than just reading—it changes the behavior of the interpreter. This is obvious with computer code where execution changes the state of the computer, but it is also true of a code of ethics (which should change the behavior of a professional) or directions to get somewhere (which should change the actions of the person following the directions so they get to the desired destination).

We can say that code is in the imperative voice—commanding the interpreter who follows the code to do certain things and not

9

other things. But text has many voices, including the imperative. Text is capable of combining voices into, for example, a dialogue like the one we're having right now.

RAMSAY: [*Long pause.*]

ROCKWELL: What's the matter? It's your turn.

RAMSAY: It says here I'm supposed to smile. I don't want to smile.

ROCKWELL: Very clever. Come on, follow the script.

RAMSAY: Perhaps we have reached the point where we no longer know what we're talking about. There's something fishy about that definition you offered. It seems to me that code is not merely a set of instructions—that it, too, has a number of rhetorical voices beyond the merely imperative. It is, after all, a language. This, in fact, may be the strongest point of affiliation—particularly since, for most programs, that language is recognizably some form of natural language (usually English). Why wouldn't code inherit some of the same rhetorical valences which natural languages possess?

ROCKWELL: But code languages really are different from human languages. For one thing, languages like Perl, Java, and C++ are simpler and more constrained than human languages like English and French. Code may be text, but it is text within a particular set of languages that have a history different from human languages. That we can read code—that code resembles English

10

text—is a feature of most computer languages. They are designed to be close to languages we know. But that doesn't change the differences in how we learn programming languages, how we use them, the contexts of production and consumption, and their grammars as compared to human languages.

RAMSAY: Yes, but a constrained language isn't less communicative by virtue of being constrained. If anything, a constrained language has the potentiality to be more communicative. Think of a haiku poem, or a sonnet—or, for that matter, a Perl poem.

ROCKWELL: Ah, but that last example is a slip of the tongue. Perl poems (not to mention ordinary poems) usually *can't* be executed. Think of the Code Poetry examples Marie-Laure Ryan showed us in the plenary she gave. Few of those examples could be executed—primarily because they don't operate according to the principles of substitution which govern coded languages. Whether code is meant for humans to interpret (as in a secret code) or computers, there is a code book or set of rules for the substitution of codewords/numbers/groups with either plain text that is humanly readable or machine instructions that can control a machine. Text is not made up of units that are expected to be substituted for something. Words do not mean something through substitution despite semiotic theories that argue that words mean because they stand in for the things to which they refer.

RAMSAY: Are you confident of this? Are you confident that what we call "mind" isn't functional—that we don't have the wetware equivalent of interpretative engines that perform a kind of computation?

ROCKWELL: [*pause.*] No, I am not confident that I know how the mind works. Cognitive science may prove I am virtually a machine. What I am confident of is my experience of the mind—or, to be more precise, my experience of consciousness, and it is as consciousness that I experience writing text and reading it. While I have no idea what the experience of executing code is like for the computer, I doubt it is like the consciousness of reading unless the computer is a Searle Chinese Room with a homunculus within following instructions.

RAMSAY: Aha, then you might agree that programming, the human activity of writing code, of which you are conscious, is as writing. If you focus on the experience of the script writer then programming is a form of writing, even if it is one intended for two types of audience, the human and computer interpreter, and not one. We do both before the computer: typing strings, cutting and pasting, drinking lots of coffee, continually re-reading what we have typed, and finally outputting it for interpreters.

ROCKWELL: Well, then we are back to the beginning. We seem to have argued ourselves into thinking of programming as writing,

which means the *Phaedrus* is relevant to a computing humanist like yourself. So now ask yourself how you would answer Phaedrus if, after his conversation with Socrates, he came back and reproached you for confusing script writing with the pursuit of truth? What would you say if he asked you to defend your text code and demonstrate its inferiority to the writing on the soul of a good face-to-face dialogue.

RAMSAY: Well, I suppose I would ask Plato why he wrote such fine dialogues with such noble ideas if writing is so unreliable a guide to wisdom? All this Plato you quote in order to improve me is transmitted through the web of writing—a web of many more than ten years.

ROCKWELL: But Plato wrote dialogues, not speeches—least of all instructions. He deliberately refrained from writing his ideas as speeches as he tells us in Letter VII.

RAMSAY: And is the dialogue any less of a script for performance? What are we doing now?

ROCKWELL: I think we are provoking questions, so, lets stop and face them.

KOVACICH: Great. That's a take.

RAMSAY: I don't know. I think we need to add in the stuff that people asked about at the end.

ROCKWELL: Like what? I don't remember.

RAMSAY: You know, Wendell asked if we weren't putting forth too narrow a definition of code and programming, since not all languages are imperative.

ROCKWELL: Hmmm, and what was the point?

RAMSAY: No, no. It was a good question. He's right that not all languages are imperative (though I suppose they all get reduced to imperative functions at the level of the machine). But frankly, when I look at some of these other paradigms, I see still more suggestive analogies with writing.

ROCKWELL: Like what?

RAMSAY: Well, take object-oriented programming, for example. Object-oriented programming is based around the notion of encapsulated objects. Each object defines some entity relevant to the space of the problem, and provides methods or "hooks" for accessing its particular behaviors. The key thing about such objects, is that they effectively hide their particular manner of carring out their assigned behaviors through what is called *encapsulation*. So, for example, if we were writing a program that processes poems, we might have a poem object with functions like `read()`, `print()`, `randomize-lines()`—perhaps `interpret()`.

ROCKWELL: Then the code is written to hide itself from other programmers. Components of the program are little black boxes, the inner workings of which are hidden from view.

14

RAMSAY: Precisely.

ROCKWELL: Well, surely that is another difference between code and writing, the written work doesn't hide itself. What you see is what you get.

RAMSAY: Of course writing conceals. Every text hides others, selectively showing an interface for the next. Writings conceal that which was not written, or that which the author erased; that which was thought but discarded, or that which the author wants to say, but can't. Texts hide their creation and history behind a story of creation. For that matter the scholarly apparatus of academic criticism hides that which which is critiqued behind paraphrase, quotation and footnotes.

ROCKWELL: But you're just stating the obvious, that any artefact hides that which it is not just by being some-thing. Encapsulation is by design, the code could work without it. To make Wendell's point you have to show me that a text could have the same effect without hiding what was not written. What, for example, could this dialogue be hiding which should be shown?

References

Abelson, H. (1996). *Structure and Interpretation of Computer Programs*, 2nd edn, MIT P, Cambridge.

American Heritage Dictionary of the English Language (2003). 4th edn, Houghton, http://dictionary.reference.com/search?q=code.

Birkerts, S. (1994). *The Gutenberg Elegies: The Fate of Reading in an Electronic Age*, Faber, Boston.

Plato (1989). *Plato: The Collected Dialogues*, Princeton UP, Princeton.

Postman, N. (1992). *Technopoly: The Surrender of Culture to Technology*, Knopf, New York.