From *ABR*, September/October 2001, Volume 22, Issue 6

Introduction: *Codework*

Alan Sondheim, Focus Editor

"Codework"—the computer stirring into the text, and the text stirring the computer. This special topic presents several reviews of the current state of a literary avant-garde concerned with the intermingling of human and machine.

"Code" can refer to just about anything that combines tokens and syntax to represent a domain. In a sense, natural language encodes the "real," gives us the ability to move in environments constantly undergoing transformation. In a narrower sense, code refers to a translation from natural language to an artificial, strictly defined one; the syntax of Morse code, for example, has no room for anomalies or fuzziness. Computer programming gener-ally requires strictly defined codes that stand in for operations that occur "deeper" in the machine. Most users work on or within graphic surfaces that are intricately connected to the programming "beneath"; they have little idea how or why their machines work.

For thousands of years, writers have, again in general, taken their tools—taken writing itself—for granted. Even Sterne and Carroll work within traditional means. The computer and Internet, however, have opened up a whole (and indefinable) world of possibilities. These range from writing itself to multimedia, and from writing-on-the-surface—traditional writing or hypertext—to texts, dynamic or static, that reflect the bones, the molecules and atoms, of programming and protocols—even the bones of the user's computer, which may be accessed by various programs. I see codework as at least one future of writing—in part, it's prosthetic, an uneasy combination of contents and structures.

Using the metaphor of a tree, codework can be placed within a very rough taxonomy as follows:

a. Works using the syntactical interplay of surface language, with reference to computer language and engagement. These works may playfully utilize programming terminology and syntax; they don't necessarily refer to specific programs. Examples include multi-media and hypertextual works—they're the leaves and bouquet of the tree, the efflorescence. I think of Mez's work in this regard, some of Antiorp's style (but see below), and some of the Internet Relay Chat jargon endemic in various chats.

Code refers to a translation from natural language to an artificial, strictly defined one.

b. Works in which submerged code has modified the surface language—with the possible representation of the code as well. Here we have the potential for continuous surface deformations. They're the tendrils and branchings of the tree, half surface and half root. Some of my own work fits here, as does the work of Ted Warnell. The language becomes increasingly unreadable at times; it's the result of a group of processes and catalysts that may or may not be reworked. (I think of Talan Memmott's work between a and b here.)

c. Works in which the submerged code is emergent content; these are both a deconstruction of the surface and of the dichotomy between the surface and the depth. I think of Antiorp's and JODI's dynamic sites for classic examples. These works are the rhizomatic roots of the tree (I recognize the botanic problem here). In order to understand what's going on, it helps to look at source code (which can be part of the content).

"C" can also refer to aleatoric or randomized work—haiku, language, or other poetry/poetic generators. Sometimes the work only appears randomized, and some times it's entirely out of control. I think of John Cayley's work here.

All of these categories move between static productions (which may or may not

be the residue, reworked residue, or simulacrum of programs and/or program output) and dynamic processes—movement on the screen, within or without the traditional window or other framework. Sometimes the computer crashes, especially with category c—and that's part of the work, part of the process.

I'm excited by all of this. It leads to vast uncharted domains (if that's still a usable term) of new and future literatures—domains that recognize the vast changes that have occurred in human/machine interaction—changes that affect the very notions of community and communality. Some of this work depends on network distribution; some of it works primarily with a lone user at his or her computer. The works themselves may often be created through collaboration: no one really knows if Antiorp/Integer/etc. is one or many people; Mez uses a pseudonym; and I work with a number of "emanants," characters who are part me, part themselves, part machine.

"Virus 2" by Alan Sondheim

This special topic presents five essays dealing with codework. Belinda Barnet writes on Ted Nelson's projects; Nelson is a pioneer in thinking about linked work, and his work is increasingly important. Beatrice Beaubien writes on Mez and Antiorp (nn / NN), presenting a text of practice and theory that opens new grounds for thinking through their work. Florian Cramer focuses on the nature of software, code, and the writing subject; the historic elements—thinking through Henry Flynt and Donald Knuth, for example—are critical to current work. Talan Memmott focuses on both the nature of codework and a number of artists/writers—Ted Warnell and Brian Lennon, among others. He focuses on inscription and elsewhere has been developing a phenomenology of codework. McKenzie Wark discusses precursors to codework as well as extended writing; his examples include Mez, JODI, Kenji Siratori, and myself.

I find these essays brilliant; they give a variety of theoretical approaches to a body of difficult work. They also extend codework itself into territories of more traditional media and the history of writing. I can only hope this introduction does them justice.

Alan Sondheim is Associate Editor of Beehive*, co-moderates the Wryting and Cybermind e-mail lists, is teaching at Florida International University, lives in Brooklyn and Miami, has been working on the Internet Text at* http://www.anu.edu.au/English/internet_txt*, was the Trace on-line writing community's second virtual writer-in-residence, and makes video/sound work on the side.*