

Code as Ritualized Poetry: The Tactics of the Transborder Immigrant Tool

Mark C. Marino <markcmarino_at_gmail_dot_com >, University of Southern California

Abstract

The Transborder Immigrant Tool is a provocative mobile phone app by the Electronic Disturbance Theater (EDT) that provides sustenance to border crossers by leading them to water and guiding them with poetry. Although the tool can be applied to any border, the chief border it has been tied to and tested on is the US-Mexico border. The EDT present the project as an artistic disruption of the tired national political theater staged at that border. The piece refocuses attention on the basic human needs of those caught in the middle of the stale and stalemated divide. For the EDT, every part of the piece participates in this disruption not merely the finished app or the poetry but the code as well. In this paper, I ask, what would it mean for the code to poetic disruption? One set of poetry for the project created by Amy Sara Carroll offers instructions for desert survival. By presenting instructions as poems, she offers one entre into reading the source code of the app as poetry. Using the methods of Critical Code Studies, I read the code of TBT in light of and as part of the poetic intervention of this complex performance.

The desert is an ecosystem with a logic of sustainability, of orientation, unique unto itself. For example, if the barrel cactus — known otherwise as the compass cactus — stockpiles moisture; it also affords direction. As clear as an arrow or a constellation, it leans south. Orient yourself by this mainstay or by flowering desert plants that, growing toward the sun, face south in the Northern Hemisphere. (Amy Sara Carroll; from the poetry of the Transborder Immigrant Tool, printed with permission of the author. [Carroll 2011])

`/* through a lot of tedious testing, I discovered that these constructors of the TextFields were throwing an IllegalArgumentException when using TextField.DECIMAL or TextField.NUMERIC constraints. The following is from the Javadoc. It seems not to contradict the use of TextField.DECIMAL or NUMERIC given that I was setting the forms to a decimal/numeric value... hmmmmm... this must be an issue in in iden implementation....*/` (Jason Najarro; from the poetry of the Transborder Immigrant Tool, printed with permission of the author. [Najarro 2007])

These epigraphs offer excerpts from two collections of poetry from the controversial artistic mobile phone application, the Transborder Immigrant Tool.^[1] The tool, or TBT^[2] as it is known, serves as a last-chance rescue application to sustain those vulnerable travelers who find themselves in hostile territory by leading them to water and sustaining them with poetry. “Transborder” could (and does) refer to any border: political or otherwise. Yet the use of “border” and “immigrant” in a project emanating from just north of the US-Mexico border, unmistakably signals engagement with

incendiary border politics that demonize the undocumented as “illegals,” as an incursion of dangerous, job-stealing invaders. This artwork inverts that narrative by marshalling empathy for the border-crosser who has already passed into the United States but who is about to die of thirst. Its tactic: drawing the audience into a ritualistic enactment of that perilous journey. However, by presenting the journey, the work does not aestheticize the undocumented as avatars for first-world observers, but instead, by reframing the journey in life-or-death terms, helps to deny the rhetorical construction of “illegals,” by recasting the travelers as immigrants in search of the most human needs: water for their bodies and poetry for their souls.

The Transborder Immigrant Tool is a mobile phone application being developed by the Electronic Disturbance Theater in residence at UC San Diego as the b.a.n.g. (bits, atoms, neurons, genes) lab. When deployed, the application, or app, will help a traveler crossing the desert to the north of the US-Mexico border, presumably on foot, to find water by means of a simple compass navigation device, aural, and haptic cues. Once the device finds a water cache nearby, the tool begins its wayfinding process, leading the traveler, likely dehydrated and disoriented, to the nearby cache. These caches have been placed in the desert by volunteer organizations, specifically Water Stations, Inc. and Border Angels, humanitarian organizations that work to fill brightly-painted barrels, labeled “agua,” with gallon jugs of water, organizations that draw volunteers from the right and left of the political spectrum in America [Stalbaum nd].

The app uses GPS information from an inexpensive Motorola phone to find the traveler’s location. Although this tool will not provide sustenance for an entire trip across the border, it does attempt to aid the traveler in what its developers refer to as the “last mile” of the journey.^[3] The traveler activates the phone in their moment of extreme dehydration, since the phone has only approximately an hour’s worth of battery charge, and after locating its position, the phone searches for nearby water caches. It is important to note that as of the writing of this essay, the TBT has not been used by undocumented immigrants dying in the desert but instead has been tested by the EDT team and has been implemented rhetorically by fans and foes alike, for whom the mere mention of the Tool stirs strong emotions.

The Transborder Immigrant Tool is not made of hyperbole and political caricatures but instead, as the epigraphs demonstrate, a series of instructions and procedures written in a calmer tone, that nonetheless draw the reader into imagined journeys. In the first epigraph, Amy Sara Carroll so beautifully intertwines survival knowledge of the desert with evocative imagery of associations. This piece is one of several poems recorded to be heard by the travelers who use the tool. The second, found in the source code of the project,^[4] is a comment written by Jason Najarro as documentation of a problem in the code and an expression of his frustration navigating the desert of java. Rather than segregate these texts as art and supporting software, the Electronic Disturbance Theater present both code and audio layers as poetry.

Just as the poems help the traveler navigate the unforgiving climate of the desert, the complex terrain of the code requires a guide to navigate the dangerous landscape of the man-made methods and the interoperating systems of both hardware and software on which the code must operate. However, rather than merely draw an analogy between the comments and Carroll’s survival poems, I would like to examine the code itself as another channel of poetry and poetic intervention, not because it resembles the structures of traditional verse as is often the case in Perl Poetry or many instances of codework, or poetry that plays upon the semantic structures and conventions of code, but rather because the code represents a text of semantic signs which when performed (either through human or machine reading) intervene in culture by creating a poetic disruption.

The poetry in many ways teaches readers the aesthetic conditions under which the code is poetry. Both poetry and code present scripts of performed instructions which act as channels of artful communication, intervening in overdetermined narratives of undocumented immigrants. In this essay, I will situate the artful communication of the Tool as a folk ritual, which when enacted, conveys valued cultural knowledge. Whether in code or poetry, these instructions for an imagined journey are embedded with cultural imagery, resonant with customs of affiliated cultures, and address a set of possible stories, or use cases, about a traveler who is about to perish while crossing the desert.^[5] To read the code, to trace its functioning as an imaginary border crosser is either sustained by water and poetry or perishes, is to engage with a counternarrative of the border that contradicts the popular rhetorics of pundits and politicians.

2

3

4

5

6

Part 1: The Poems

The TBT project grew out of the unique ensemble that makes the Electronic Disturbance Theater. It began in 2007 as an extension of EDT-member Brett Stalbaum's Walkingtoolsgpx API, a library of tools for use in art projects involving GPS-navigated walks, often in rural or even dangerous areas. UCSD professor, Ricardo Dominguez, proposed to his EDT collaborator that they consider applying these tools to the US-Mexico Border, and the Transborder Tools for Immigrants (as the project was originally called) was born, though not before the EDT applied for funding both to purchase phones and to develop a poetry component of the project to be written by EDT-member Amy Sara Carroll. As the project progressed, the collective developed all elements, including the poetry and code, in conversation with one another, according to Carroll [The Electronic Disturbance Theater 2011] The group began the development in earnest together with the other members of the EDT, Elle Mehrmand and micha cárdenas, and a UCSD undergraduate, Jason Najarro, who over the summer of 2008 wrote the core code for the Tool together with Stalbaum. Even while in-development, the tool they have created has caused quite a disturbance.

7

Within a few months of the publication of articles about TBT in 2009 in Vice Magazine (Dunbar) and then the geek culture link site Boing Boing, the University of California received a letter from Congressman Duncan Hunter and two others. Hunter “found the project to be a poor use of taxpayer money, particularly doing a recession” [Miller 2010]. As Fox News gave the story national coverage, the University of California began an audit into the project as several other investigations began to pursue Dominguez. Between op-ed pieces and a featured spot on the Glenn Beck show, the project grew in infamy, even as the production of the project was delayed, provoking violent emotions even before the app had ever been used (Morlan).

8

The Transborder Immigrant Tool exemplifies Rita Raley's concept of “tactical media,”^[6] whose “critical rationale” is “disturbance” [Raley 2009, 19]. She explains, “In its most expansive articulation, tactical media signifies the intervention and disruption of a dominant semiotic regime, the temporary creation of a situation in which signs, messages, and narratives are set into play and critical thinking becomes possible” [Raley 2009, 19]. In her book of that title, Raley posits tactical media as a response to “a shift in the nature of power, which has removed itself from the streets and become nomadic” [Raley 2009, 1], a notion derived from the Critical Art Ensemble, one of the major creative influences of EDT and of which Dominguez was a member.

9

Again, disruption is at the heart of these tactics. For example, by focusing the application not on the overcoded moment when the *illegal immigrant* crosses the border but on the moment the border crosser is dying in the desert, the project disrupts the contemporary neoliberal narratives about the border. As Raley explains about another EDT project,

10

The aim is not to theorize liminality but to force a rupture in the binaries of interiority and exteriority, here and there, native and alien, friend and enemy. The radical dichotomies integral to the war on terror — “you're either with us or against us” — find their counterpart in art practices that themselves depend on the solidarity of the “we” against the “them.” A fence has been built, binaries constructed, and these artists intend to overturn them. [Raley 2009, 50].

This project overturns the border binary by rephrasing the question: “What should be done about those national subjects who cross illegally?” as “What should be done about those humans who have already crossed but are now about to die in the desert?” In this way, the Transborder Immigrant Tool essentially leaps over the border fence and thus escapes the pageant that plays politics with the poor as it resettles in the realm of life and death and humanitarian aid. However, the app not only can lead a person to water, it can also deliver aid in the form of poetry. Here is another disruptive tactic of the Transborder Immigrant Tool, breaking the categorization of the app as a mere tool of physical survival while it also disrupts an image of the border crosser as a laborer for whom poetry would be at best a distraction. The EDT's message is clear: this Tool is an aesthetic object, resonating through artistic interventions even while providing for fundamental physical needs.

11

The poetry for the app was written by Carroll, though other members of the Electronic Disturbance Theater have been contributing to the larger corpus of poetry which has been shared in other contexts, such as museum installations.

12

Carroll describes the poetry as very different from her other work but suited to the occasion (email correspondence), describing her process of developing them after reading piles of survival guides and the migration account of Luis Alberto Urrea in *The Devil's Highway* [Carroll 2012].

Carroll situates these poems within the larger context of “conceptual writing,” as laid out by Craig Dworkin, Vanessa Place, and Robert Fitterman. Dworkin identified this genre, drawing upon the works of Kenneth Goldsmith, as “non-expressive poetry,” akin to conceptual art, “in which the idea cannot be separated from the writing itself: in which the instance of writing is inextricably intertwined with the idea of Writing: the material practice of *écriture*” [Dworkin 1993]. However, Carroll notes that the piece moves away from the aesthetic-focus of North American conceptual writing toward the Latin American tradition, which embraces a more explicitly political mission.^[7] A full review of conceptual writing is beyond the scope of this article. Suffice it to say, by gesturing toward conceptual writing, Carroll situates the poetic in a realm that moves beyond surface characteristics, where formal conceits and formalized procedures become central to how the writing is read – a genre in which writing process and computational algorithm become much more compatible bedfellows.

13

That is not to say that Carroll has written code poems or codework. Carroll’s poetry engages with the aesthetics of instructions, of survival knowledge, derived from folk wisdom more than the episteme of scientific knowledge epitomized by the mobile phone that delivers the app. The poems themselves are individual meditations on some aspect of the desert, particularly the Anza-Borrego desert in San Diego County, as each poem combines imagery of the desert with tips on survival in that trying terrain. In the poem that serves as one of the epigraphs to this paper, for example, the reader, or more properly the listener, will encounter information about the barrel cactus, which suggest how it can be used both for physical sustenance and direction. Other poems offer advice on how to cope with and manage exposure to the sun, dehydration, and desert snakes, while conserving energy in the extreme heat. The nature of these poems reframes the tool with a larger tradition of ritualized folk knowledge.

14

First, though situated as poetry, these passages offer practical instructions, survival guides, embellished with poetic flourish. Folklorist Tok Thompson offers the parallel of “leaves of three let it be” as a sign of the poetry of folk wisdom [Thompson 2011]. Similarly, these poems convey knowledge of the land, its dangers, its hopeful sustenance, knowledge derived from the land, delivered in a deceptively plain-spoken style. These instructional poems thus evoke forms of artistry that transmit communal knowledge, like folk tales, aphorisms, memorable for their images and formulations, yet crucial for their practical information. They deliver poetic imagery with the aesthetic of instruction.

15

Reading the poetry as instructions with deep communal ties, or instructions that lead into the folk, situates the poems as the mirror image of the code, a set of instructions which also bear the deep imprint of cultural practice. Or rather, if the poetry seemed an unlikely place to find practical instruction, the code strikes many as an odd place to find culture. However, as the poetry takes up these practical formulations it outlines the nature of a kind of folklore, how the practical instructions bear the communal practice and wisdom.

16

Second, the poems are performed. They are read aloud (in multiple languages) through recordings on the phone. Although the text I quote in this paper has been copied from a digital text file shared with me by Carroll, those travelers who encounter the poems will hear them played through the phone at a random moment while using the Tool. The poems have been recorded by cárdenas, Merhmand, and others, and stored as wav files to be played on the mobile phones as part of a portion of a set of use cases. The poems that play to them are not part of a collection, an anthology, to be perused or thumbed through, but part of this process. That process is not merely accessible to the person who is dying in the desert but it is also accessible to anyone who traces their way through the code (or code repository) either using the machinery of the phone (or the software equivalent), or by reading through the code. The poetry then offers knowledge situated in a performance that is part of a process.

17

But the poetry itself also suggests folk ritual, as it offers knowledge or advice that has been embroidered with rich semantic inflections. Consider again, the poem from the epigraph. The simplest version of its contents is “the barrel compass stores water and points south.” However, Carroll frames this information with the metaphor of “an ecosystem with a logic of sustainability, of orientation.” Of course, the phrase “logic...of orientation” speaks both to the poetry but

18

also to the tool itself, which now by extension is grounded in the deeper nature of the desert. Evoking the basic compass wheel interface of the Tool itself, Carroll refers to the barrel cactus in its alternate name, “compass cactus.” Abruptly, the sentence switches its frame with the metaphor of “stockpiles,” an industrial term more often used not in survival but in accumulations of destructive materials. And again, she shifts registers with “affords,” a term with deep resonance in the realm of tool design, programming. The end of the poem entangles or replaces the tips for survival knowledge into the larger realm of geography and contested borders, as the piece encourages the traveler to orient himself or herself by facing “south in the Northern Hemisphere,” which at once evokes a sense of human divisions of the land in the context of a world view into which nation-states and what Dominguez calls “the aesthetic fiction of the border” dissolve [Dislocative 2010b]. Carroll’s seemingly plain-spoken directions call forth the conflicts of world views, pulling on metaphors from production, war, and design, while re-orientating the reader toward land and country.

It is perhaps only in the realm of science that we entertain a notion of cultural knowledge devoid of cultural dispositions.. Without rehearsing the entire epistemes of Foucault, it is worth noting that scientific knowledge is culturally situated and constructed and that the long shadow of world views colors or frames all descriptions of empirical phenomena. The genre of “directions” or “instructions” become emblematic of knowledge that has been systematized, seemingly stripped of unnecessary verbal trappings, especially when what is being described is a naturally occurring process. It is easy to conceive of computer source code in this context: a series of instructions (although not everything is an instruction) devoid of cultural inflection, pure process.

Carroll’s poetic instruction in the Transborder Immigrant Tool, however, suggests ways in which instructions become more than mere directions. Her poems draw to the forefront the extent to which knowledge, even knowledge of survival, when communicated is, by necessity, situated in other kinds of cultural frames of reference – that knowledge is always cultural knowledge and instructions are always something more. Her work calls to mind other forms of practice that more obviously mix instruction with story, with song, with art.

Yet Carroll claims that the code of the tool is the poetry of the project. Such a claim might be more disturbing than anything else called forth by the project, and yet given the context of conceptual writing, the code, as a manifestation of the logic and loglines of the art project, becomes poetry not just through that very framing but also by the very instructional nature of the poems it delivers.

Part 2: The Code

[8]

To read the poetry, is to imagine a traveler in her hour of need receiving just-in-time guidance and sustenance. To read through the code of the Transborder Immigrant Tool is to stage a ritual in which mythical travelers cross the desert through the software.^[9] On the surface of the comparison, these traversals seem inconsonant, since one involves physical movement and the other an act of the imagination of execution. I would argue that since the act of reading code requires models of materially tied processes, models, even black-boxed models,^[10] of the effects of the executed translation of the code on software and hardware, it is not necessary for someone to be physically moving to be enacting this journey.

The code of the Transborder Immigrant Tool is set to function in a particular set of use cases of a traveler lost in the desert in desperate straits. To read the code is to implement the code, to imagine the realization of that code, the software produced by it, and consequently the desert and the desert traveler as well as the water to which the software leads. To read the code is to enact the ritual of the code which carries with it communal knowledge and a communal history. Such an enactment is not meant to trivialize or aestheticize the real lives of those who could benefit from using the tool but instead to meditate not on their political representations but rather on their (potential) lived conditions by tracing through a procedure designed to rescue them.

Code is routinely conceptualized as abstracted procedure. However, for this paper, I am setting that conventional term aside for a more culturally situated formulation. “Procedure” calls to mind a series of steps aimed at a goal; “ritual” is a performance aimed at cultural connectivity, at instilling community, of signifying allegiance, of remembering. The code

like the text of the oral performance is more of a transcript of a performance to be realized in the mind of the reader or by the processes of the computer. It bears signs of its performance as well as its history. Which is not to say that code does not have an end in mind. The Transborder Immigrant Tool does produce the app and govern the interactive experience of the app. I am focusing rather on the human reading of the code. Thus, I would argue that to engage with the Transborder Immigrant Tool through the code is to engage in a ritualistic imaginary performance of the software, a speculative deployment, an engagement that has many parallels with folk practice of ritual, song, and tale.

The act of reading code then involves a willful attempt to process like the computer, to emulate the system that will be operated on by the performative, automatic, triggers of the code. The notion of literacy, therefore, in the sense of reading literacy, is insufficient for the act of reading code. As in processing textual symbols, reading code does require an understanding of the connotation of the instructions. However, connotation and electrical effects are not the same. Andrea diSessa in her book *Changing Minds: Computers, Learning, and Literacy*, offers the notion of “Material Intelligence” to substitute for literacy, meaning “intelligence achieved cooperatively with external materials” [diSessa 2011, 5]. Because of the specifications of hardware and software, including operating systems, it is not enough to say that reading code requires the same kinds of knowledge of the interdependent corollaries and theorems as in the case of mathematics, though as diSessa notes these both require similar skills. Annette Vee offers the term proceduracy “for the literacy associated with computer programming, is the ability to break down complex processes into smaller procedures and express them explicitly enough to be read by a computer” [Hunter 2009]. Reading code requires establishing an adequate representation of the state of the software (and at times the hardware) at any given time in the program. As I have mentioned elsewhere, this work can often function as a threshold paradigm that conflicts with the act of interpretation, which looks to connotation rather than this highly material form of denotation.

25

To examine the code of this piece, I will use several of the methodologies of Critical Code Studies, an area of study I have been collaborating in since 2006. Critical Code Studies is the application of humanities-style hermeneutics to the interpretation of the extra-functional significance of computer source code. It is a form of media-specific analysis that uses the source code as an entry point for discussion of technoculture. Code readings have examined databases, computer games, works of electronic literature, and even a program that is a single line of code long. The goal is to develop a more detailed understanding of encoded objects through engagement with the unique, historically and materially situated signs of their code.

26

The Transborder Immigrant Tool was written by then-undergraduate Jason Najarro and UC San Diego instructor Brett Stalbaum in Java, specifically Java 2 Micro Edition (J2ME) for the IDEN platform on the Motorola i455, a comparatively inexpensive mobile phone. Java, originally developed by Sun Microsystems, has been an open source programming language since 2007, undergoing development from a worldwide community of programmers. Like Java, the Tool itself is offered with a GNU Affero General Public License, which makes it available for others to continue or modify.

27

The Tool also makes use of pre-existing libraries, namely those of Java and J2ME and Walkingtools, developed by Stalbaum and others. Stalbaum has described these tools as an open platform he has been developing for use in art and education projects [Stalbaum nd].

28

Attribution and Affiliation in Code

A flag salute, a kiss on a cheek, a communal dance, these ritualistic conventions create a sense of connection even as they serve as sorting methods, as shibboleths, as barrier to entry, a border between inside and outside the community of knowledge. In this way ritualized forms signal affiliation.

29

Code is similarly marked by such affiliation, and the Transborder Immigrant Tool establishes its community with the open source code movement through its choice of programming language, its licensing agreements, and, of course, by releasing the code on the SourceForge site. Bruce Perens, who developed an early definition for “open source,” wrote in 1999 that it is a specification for licenses that ensures: “The right to make copies of the program,” “The right to have access to the software’s source code,” and “The right to make improvements to the program” [Perens 1999]. In that seminal article and his definition he promoted a new model for software development. The Transborder Immigrant Tool is a piece of folklore that circulates in affiliation with the open source folk and all the attendant ethos of collaborative

30

development and sharing of resources.

Consider the header code in the DowsingCompass holds many such signifiers of connection: The comments at the top of the code begin the file by locating this code within the larger Walkingtools project, an platform for developing apps such as TBT. The second line announces a copyright even as it precedes the explanation that the software is free and can be redistributed freely. The text of that copyright, like many portions of the code, serves a performative function that by applying the GNU Affero General Public License, signals and establishes the projects links with the Open Source movement.

31

```
/* WalkingtoolsGpx: XML, APIs, and Apps for Walking Artists
Copyright (C) 2010 Walkingtools project/B.A.N.G. Lab UCSD
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as
published by the Free Software Foundation, either version 3 of the
License, or (at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Affero General Public License for more details.
You should have received a copy of the GNU Affero General Public
license along with this program. If not, see <http://www.gnu.org/licenses/>.
```

This license text presents another instance where language is operating as functional, performative utterances even as it signals affiliation. Nor is this license text a unique expression of the programmers. Instead it is boilerplate, or rather a scripted part of the ritual, used by many across the community. Yet even as the project reaches for the freedom of the open source community, it offers a nod to the legal regime under which it would fall victim if it event “implied” a warranty of “FITNESS FOR A PARTICULAR PURPOSE.”

32

The Open Source community becomes a folk through its licenses, which have become precedent for the Creative Commons licenses that have so shaped creative symbolic production and circulation on the Internet. At the turn of the millennium, Rick Borovoy et al. (2001) described a process they called “Folk Computing,” “an approach for using technology to support co-present community building inspired by the concept of folklore” [Borovoy 2001, 1]. This notion, born of the MIT Media Lab, becomes “Folk Programming.” In a presentation entitled “Forking Encouraged: Folk Programming, Open Source, and Social Software Development,” Kirrily Robert, Jason Douglas, and Yoz Grahame use “folk programming” as a way of naming contemporary open source software development. In both cases the open source programming community becomes the site of a folk, since the emphasis is on collective development.

33

Of course, that is not the only form of affiliation. The choice of a programming language represents a source of affiliation, linking this project to the ethos of a particular programming language as well as its paradigms. Also, the application affiliates itself by making use of pre-existing libraries, in this case various Java and J2ME libraries, and more particularly, the Walkingtools, an open API developed by Stalbaum in collaboration with others to create a platform or infrastructure for projects just as this one. These libraries serve as a necessary groundwork for the application, while situating the Tool itself in a clan or network of software applications. The code calls these libraries through import statements:

34

```
import edu.ucsd.calit2.TransborderTool.international.*
```

The name bears the affiliation of the code to its source, as a file name and as an institution, calit2 at UC San Diego. The naming reflects a file structure, that following Java conventions is a kind of reverse URL. As a result, the code acknowledges the institutional context that fostered its growth and would later investigate its production. At the same time, the mere act of following that convention is itself an act of affiliation to the larger community of Java programmers, the folk to whom this lore will be legible.

35

In these ways, the folklore of code points to its community.

36

The Lore of the Comments

Jeremy Douglass argues, "Comments are part of a vernacular, with their own folk traditions and lore"(Douglass, Jeremy) . I would push this a step further by saying that comments themselves are lore, and those in the Transborder Immigrant Tool take a form that parallels the embellished instructions of the poetry. Jason Najarro's commentary in the Transborder Immigrant Tool code possesses much of this quality found in Amy Sarah Carroll's writing. Consider this passage from the TBMIDlet.Java code in which Najarro seems to be leaving trail marks and warnings.

37

After writing the comment included in the epigraph above, Najarro quotes the following passage from the IDEN documentation,

38

```
191. "Some constraints, such as DECIMAL, require the implementation to
192. perform syntactic validation of the contents of the text object.
193. The syntax checking is performed on the actual contents of the text
194. object, which may differ from the displayed contents as described
195. above. Syntax checking is performed on the initial contents passed
196. to the constructors, and it is also enforced for all method calls
197. that affect the contents of the text object. The methods and
198. constructors throw IllegalArgumentException if they would result
199. in the contents of the text object not conforming to the required
200. syntax."
```

The code is authorless yet authoritative, like an invocation of holy writ or communal law, and yet this text is something much more mundane and patently incomplete.

39

Through Najarro's comment, his notes on his own frustration, the code becomes a document of process, spoken in first person to an audience that is trying to make their way through the code. The text takes the form of an oral remark, as he interrupts himself with "hmmmm" and then offers speculation. Then, he leaves a bit of text from the manual, the document that was supposed to tell him how to interact with the IDEN system as a sign of the well that was empty.

40

Throughout this well-documented code, Najarro's voice returns, translating the functioning of the code into straight forward imperatives, for example before running the `getNearbyWaypoints`, Najarro writes, "Offer any available help to user" [Najarro 2007, 429]. Yet the voice is not addressing a listener but paraphrasing the activity of the code. In another passage he writes, "we are now returning from a bad signal" [Najarro 2007, 424]. The royal "we" here stands in for the code and perhaps also its programmers. The comments use the modal helping verb "may" in a manner that establishes the uncertainty and indeterminacy, as in this example: "in these cases the user may still be able to navigate with a magnetic compass" [Najarro 2007, 408]. Again, such a state of uncertainty or potential is the state of the unexecuted program, paralleling the advice of the prose poems, where all instructions are offered with contingencies.

41

This is not to suggest that the comments were all written by one person or in one voice. Consider the text of the `TBgpxParser` file which explains:

42

```
/* Completely rebuilt this part on may 10th, 2008, in order to make the
class
* more robust. Before it assumed that certain elements would be contained in
* a particular order, without any intervening xml tags that might be added
* by various applications. (Such as G7towin, which produces gpx files that
* this class threw up on.) I tried to make it more robust by looking for
tags
* and accepting their data if they contained "wpt" data, and just ignoring
* anything else it finds. Brett
*/
```

A class "throwing up" on gpx files does not appear to be the standard procedure for Java. In this passage, this comment

43

returns to the decidedly personal record of the designer's log rather than paraphrasing what the code does. As a further sign of documentation to aid future revisions to the code, Stalbaum even signs this note. In another comment, Brett's informality seems to return:

```
54 ...[the software] should now handle any gpx file containing
55 * waypoints, regardless of other junk in the file.
```

The many voices of the commentary represent more than the multiple authorship but also the heteroglossia of the code, to use Bakhtin's notion,^[11] the many genres and styles within even one set of comments composed by the same programmer. While Bakhtin spoke of novels, these code files are full of narratives, accumulated fragments, even some auto-generated comments,^[12] organized around the particular operational patterns of the instructions.

44

Walking through the code:

To read the code is to engage in what Dominguez calls a "speculative deployment," or imagined execution, of the code [Dislocative Media 2010a]. To trace the code is to test a series of potential use cases, imagining a user, here a mythological border crosser who is about to die. (Note that aside from tests by the EDT, there have only been speculative deployments so far)^[13]. So to read this code is to travel with that user across the desert, trying to survive by using the tool.^[14] The traveler is left in the state of Schrödinger's famous cat, straddling or toggling between the states of life and death. To aestheticize the plight of the immigrant would be exploitative, but I would argue that since the tool has circulated far in advance of any implementation that the Tool is a rhetorical device even more than it is a survival mechanism. To participate in the ritual of the code is to move with that user, to travel with them, to engage in the subject positions inscribed in the code, which has been framed in a certain logic about the border. A walk through of the code will enact this ritual, while no doubt creating another "text" of the folklore that is the Transborder Immigrant Tool.

45

At the SourceForge code repository, a reader will first encounter the five main Java files of TBT: Dowsingcompass, Dowsingcompasslistener, TBCoordinates, TBgpxParser,^[15] and TBMidlet. Midlet is the name for MIDP (Mobile Information Device Profile) applications designed to be run on small devices, particularly mobile phones. The "let" suffix follows the naming convention of servlets and applets. The primary file, then, is the TBMidlet file.^[16]

46

The first part of the code initializes the variables, setting them to their default states, most of which are null, including the variables that are key to locating the water, nearbyWPList (nearby Waypoint List) and targetList. Six poem names are loaded in as well as the names of the sound files. After this initialization, the code presents the constructor for a TransBorderMidlet:

47

```
public TBMIDlet () {
```

In Java, a "constructor" creates an instance of the class (known as instantiation), in other words, a new object, the application itself.^[17] "Public" means that any file can call that function.

48

Before it can help the traveler navigate, the system needs to know where water is, so it runs TBgpxParser as a thread. A thread is a type of background process or subroutine. A commented-out line (or a line of code which will be "ignored" by machine readers) shows how one implementation of the tool loaded the coordinates from a website.

49

```
// private String URL =
"http://internetjunkee.com/transborder/GPScourseFinal.gpx";
```

In an operational version, according to Stalbaum, the coordinates would be pre-loaded onto the phone rather than posting them for anyone to find on the Internet. Stalbaum said this choice was based on the sensitivity of the water cache data and the project's work to keep it private – not because the water stations themselves are hidden, sitting just off road in bright blue barrels, but because an opponent of the project might make ill use of the entire list of coordinates [Stalbaum nd]. As a consequence of this vulnerability, the application does not (and in fact cannot^[18]) download the files but will be preloaded with them. This commented-out line of code serves as a reminder of yet another potential threat to the survival of the immigrant, those who wish to sabotage the water stations.

Having loaded the data, the software performs a “freshness test” to see if the location data has expired:

```
301. // make sure the data is not expired
302. if (isExpired) {
303. display.setCurrent(expired);
304. display.vibrate(1000);
305. playAudioFile("expired.wav", true);
306. } else if (expireWarning) {
307. display.setCurrent(expirationWarning, tbDowsingCompass);
308. display.vibrate(1000);
309. playAudioFile("expiration.wav", true);
```

If the data is expired or the warning is already set, the phone vibrates and plays the expiration warning.

```
310. } else if (startupAlert) { //first time only
311. startupAlert = false;
312. display.setCurrent(startUpDisplay, tbDowsingCompass);
313. display.vibrate(1000);
314. playAudioFile("startup.wav", true);
315. } else { // we are good to go
316. display.setCurrent(tbDowsingCompass);
317. }
```

Otherwise, if the app has just been launched, it plays its startup audio. The expression of the process that checks the expiration date seems rife with the urgency, finality.

```
126. if (exp <= System.currentTimeMillis()) {
127. expired = new Alert(translation.translate("Data expired"),
128. translation.translate("The data is expired, TBTool is not safe to use."),
129. errorImage,
130. AlertType.ERROR);
131. expired.setTimeout(Alert.FOREVER);
132. expired.addCommand(exit);
133. expired.addCommand(ignore);
134. expired.setCommandListener(this);
135. isExpired = true;
```

Again, line 126 checks to see if the expiration date was on or before the current date. If it is, then a new alert is created. Lines 132-133 create options for response including an “Exit” and “Ignore” option.

In this code snippet, the reader encounters the first scenario in which the imagined user would likely die. Reading the code as a kind of procedural narrative creates the sense of two distinct tonal registers. Notice how the more unemotional, mathematical aspects of the code (the “if (exp <=)”) alternate with a more loaded natural language of the functions, objects, and data. The lines “AlertType.ERROR” and “expired.setTimeout (Alert.FOREVER);” indicate the finality of this state since no additional waypoint information is available (TBMIDlet 127, 128). Reading these lines of code, excerpted from their coding environment, it is difficult not to notice the foreboding in the all-caps ERROR and FOREVER or to ignore the terrible resonance that builds with each repeated “expired.” However, those dramatic elements alternate with lines that seem much more understated. The simple variable assignment “isExpired = true” would most likely spell death for the traveler. In this way, dual registers sound out in the code, one that more directly announces the danger and another that, though delivering equally grave news, proceeds with a bare mathematical calm.

55

The second such scenario occurs immediately after in the code.

56

```
157. noNearbyWaypoints = new Alert(translation.translate("No Nearby Points"),
```

Translated into the traveler’s preferred language, this alert plays for 10000 milliseconds, possibly the last alert the person will hear. Such a statement sounds maudlin, but it is important to remember that this code is not merely a system of causes and effects, of calculations, but a representation of potential real-world moments with dire consequences. Like Carroll’s poetry, the code conveys these moments of imminent death without using the conventions of a protagonist, drawn-out scenes, or other devices that typically cue affect, requiring instead that the reader realize the potential peril while processing these instructions. The finality of this scenario does not come across with the sensationalism of 24-hour news but instead with a kind of muted affect, the impact only settling in once the reader has traced through the implications by tracing through the consequences of the code.

57

Of Witching Sticks and GPS

Although the Transborder Immigrant Tool relies on high-tech navigation equipment, the story of the code does not champion the potential for a technological rescue but instead reverts to folk knowledge, material bodies, and natural environs. While both kinds of scientific knowledge might be considered “folk” knowledge, the Transborder Immigrant Tool draws upon the wisdom of Death Valley rather than Silicon Valley.

58

While death awaits the traveler who has expired data or cannot find a waypoint, in other versions of the scenario, the traveler will find a destination to navigate towards. By default, the system will set the cache as the location, but the user can manually choose a destination, including a beacon or a city. The Tool searches for a nearby water station by using DowsingCompassListener, an extension of a Walkingtools method called NavigatorListener.

59

```
205. tbDowsingCompass.addNavigatorListener(this);  
206.  
207. targetList = new List(translation.translate("Select a Target"),  
List.IMPLICIT);
```

When the user is facing a water cache, the software calls a witchingEvent, a method which begins the primary navigation. This metaphor of “witching” or “dowsing” is part of the framework of the tool, yet it is not an explicit part of the project. Rather, it was added by Jason Najarro as he was programming. Elsewhere I discuss in detail the significance of such a metaphor.^[19] Suffice it to say, introducing this folk practice as a conceptual metaphor of the Transborder Immigrant Tool frames this set of methods to a pre-digital cultural practice. The high-tech communication

60

and global positioning device is turned into the stick with its innate connection to water. In a `witchingEvent`, the `Tool` begins wayfinding, leading the person to the water cache through a set of audio and haptic cues. If the immigrant is fortunate enough to reach the water, the code celebrates:

```
212. waypointAheadAlert = new Alert (translation.translate("Site Ahead!"))
221. arrivedAlert= new Alert(translation.translate("Arrived at Site"))
```

However, toward the end of `TBMidlet`, the code raises the specter of another unfortunate outcome. If the traveler stops, the `Midlet` plays a message to encourage the person to walk:

```
554. If (isMoving) { // updated to moving
555. nearbyWPList =
556. tbDowsingCompass.getNearbyWaypoints (SEARCH_DISTANCE); //so update nearby
point
557. }else { //updated nt moving
display.vibrate(200).
558. If (moveWarningEnervator % 5 == 0) { //only play this file ~ every 5th time
559. playAudioFile("move.wav", false); //the "move for compass message can be too
frequent.
```

The `moveWarningEnervator` is meant to push on the immigrant who may have stopped from exhaustion or perhaps disorientation. The vibrations and the audio output were built around Najarro's research into the state of a dehydrated person. Here is another occasion in the development of this project that demonstrates its primary focus on *servicing* a real desperate human need, rather than merely representing that need. Reflecting on the mental state of the traveler, Najarro realized that their interface must be simple enough to communicate to a person about to pass out or worse [Stalbaum nd]. Similarly, as the traveler nears the water station, the duration of the periodic vibration increase, just as a dowsing stick would respond to its approach of water in the ground.

To read this code, by my measure, is to emulate the functioning of the code in a use-case. The reader must imagine the effects on the code of a water station found nearby or none found. In this way, that reading experience is likely to be multiple, but it cannot be vague. In other words, in order to process the code as the machine does, the reader cannot avoid the question of whether or not the traveler finds a water cache, but instead in each use-case must decide whether or not water is found and, consequently, whether the border crosser lives or dies.

Poetry in Potential

Just as the code presents the use-case as instructions suspended around potential states, so too does Carroll's poetry. A second poem demonstrates this combination of instructional imperative, poetry, and potentiality: "Heat cramps, relatively mild, signal dehydration and loss of sodium. *Drink water, rest in the shade; seek water at twilight.* Heat illness is an injury whose symptoms include fatigue, dizziness, fainting, nausea, vomiting. Redux: *Drink water, rest in the shade; seek water at twilight.* Heat exhaustion produces sweating, clammy skin, increased pulse and respiration rate, weakness, more fainting, nausea and vomiting. *STOP. The choices from now on you make will dictate whether you live or die.* Heatstroke happens when a person pushes on despite heat exhaustion. Trauma ensues — physical collapse, loss of consciousness, rapid pulse and respiration, a skyrocketing body temperature, severe disorientation, impaired motor skills, involuntary urination, dilated pupils. As heatstroke progresses, you will experience chest and arm pain, convulse, go into a coma. You will not be equipped to deal with these symptoms as they present themselves. *Call 9-1-1 or 0-6-6 beforehand.*"

Similar to the code that handled the expired data, this poem offers the reader another pair of voices: one delivering

data, medical information in a dry, calm tone, the other imperatives, whose italics underscore their urgency and alarm. The passage begins with the plain tone of a definition followed by instructions on how to combat. The symptoms list grows followed by a repetition of the instructions, “drink...rest...seek water a twilight,” now a refrain. For those who encounter the art project, as opposed to those using the system to survive, the instructions are like the code: something that we read as intended for someone or something else to put into action. The second symptoms list is repeated but the pattern is interrupted by an alert: “Stop,” followed by a much more stark warning. The symptoms list resumes with elaboration, embellished with images. Through anaphora, Carroll intensifies the impact of the “heat.” The final list marks symptoms the person will not be able to react to or recover from, followed by a command. This alternation of the symptoms and instructions act as alternating threads or subroutines, the one set to draw out the set of effects of hydration, the other instructions on what to do next. Yet they are not unrelated. The symptoms serve as signs to motivate the responses, like a series of if-then statements. They proceed with increasing severity on a path to dehydration even while they are instructing on how to respond to it. The instructions, the direct address, serve as the moveWarningEnervator, to push the perishing traveler to action.

The rhetoric of folk wisdom, whether in rituals or prophesies, delivers its instructions with conditions for execution and possible scenarios for outcomes, in ways analogous to the code. In cautionary stories and warning axioms, rules and rhymes, legends and prayers, folk wisdom travels through a set of admonitions that, like the code, carries with it the outcomes when its conditions are not met. The code and poetry of this piece deliver the wisdom of those survival procedures in tones that alternate between calm, methodological instructions and descriptions, on the one hand, and alarming alerts and pressing imperatives, on the other. The juxtaposition of those tones, as well as the interplay between the poetry and the code layers, the way they frame one another, disrupt any singular narrative of the transborder traveler’s experience, offering instead a journey that proceeds with urgency and yet inevitability, inevitability and yet possibility, leading either to death or to survival.

66

At the same time, both the code and the poetry defuse the political Molotov cocktail of the border by situating the land not in the context of political boundaries but in a survival narrative. Though we have read the code as deployed by an imaginary undocumented immigrant, neither the poetry nor the code make mention of the traveler’s national identity, using TB in the naming of files and methods throughout the code to inscribe “transborder” across the project in place of “undocumented” or “illegal.” The criminalization authored by the State is hence supplanted by the moniker of movement across a political fiction.

67

Both the code and the poetry are part of rituals, part of this instantiation of the Transborder Immigrant Tool. They are performed in the execution of the Tool, but this execution has been mostly an imaginary activity. Most people who encounter this tool have not been dying in the desert. Rather, those who speak of the software and its applications engage in speculative deployments. Some deploy the project only partially based on little more than the publicity videos or snippets of poetry, leading typically to reactionary responses. However, tracing carefully through the instructions in the code and the poetry, the reader has the opportunity to imagine the potential outcomes for the traveler and more fully consider the implications separate from the hysterical political rhetoric. Like traveling a labyrinth or walking the Stations of the Cross, pursuing the processes of the Transborder Immigrant Tool requires faithfully following set paths that others have and will continue to pursue as a form of meditation.

68

To situate code as ritual is to acknowledge its place in the transmission and production of culture itself. In *Ethomimesis*, anthropologist Robert Cantwell writes of the place of ritual in culture, arguing, “Whatever the ontological status of culture as such, it must be embodied, enacted, performed, represented and reproduced in order to have any social reality” [Cantwell 1993]. The code and the poetry of the Transborder Immigrant Tool present a site of this performance, the enactment of culture through the lore of travelers who live or die while making their way through a desert that they must learn to read. In reading the code, we cross and cross back that same desert, guided by the wayfinding of the comments and advice, encountering and becoming a folk attentive not to the political spectacle of national border debates but to the potential loss of life and the conditions of sustenance for our fellow travelers

69

Notes

[1] This essay is published in parallel with a further exploration of the TBT online at <http://scalar.usc.edu/nehvectors/border-codes/> where readers will find the complete set of code as well additional interviews, background readings, and analysis. *Border Codes* was developed as part of a 2011 NEH Vectors Summer Institute.

[2] Technically, the initials of the project at TIT, and the metaphor of the life giving, maternal body is by no means a coincidence; however, in the hands of critics of the piece, the initials became a weapon to harass particularly the women of the Electronic Disturbance Theater. For more on those events, see <http://scalar.usc.edu/nehvectors/border-codes/>

[3] Virginia Kuhn has pointed out to me that “last mile” is a concept most commonly used to refer to the final distance between existing lines of fiber optic cable and home Internet users.

[4] Brett Stalbaum of EDT made the code available via Source Forge as of the 2010 Critical Code Studies at USC conference.

[5] Although the TBT has never been used by immigrants, it does not engage with the misfortunes of immigrants merely aesthetically. Brett Stalbaum has said that he made sure when developing TBT that he could trust his life with it. Equally instructive is the volunteer work EDT has done helping to refill the barrels to which the tool navigates. [Stalbaum nd]

[6] In her book of that title, Raley traces the term “tactical” and its emphasis on disruption, via Geert Lovink, to the Next Five Minutes (N5M) held in Amsterdam in 1993 [Raley 2009, 6].

[7] See Luis Camnitzer’s *Conceptualism in Latin America* for an insightful overview .

[8] The main code files for the Transborder Immigrant Tool are explored in depth in *Border Codes* (<http://scalar.usc.edu/nehvectors/border-codes/>)

[9] Though there are many ways to read any code object, including many partial readings, here I am discussing a particular reading path through the project to trace its functionality from startup to shutdown.

[10] In programming discourse, it is common to use the word “black box” to describe a process whose input and output is known but whose specific operations are not analyzed.

[11] See “Discourse in the Novel” in [Bakhtin 1982].

[12] Line 470 has the comment “//TODO Auto-generated method stub” which was most likely generated by Netbeans, the development environment that the EDT used to create the code.

[13] The tool has been used in the context of performances, for example, when the EDT used it to cross from the United States into Mexico.

[14] In this section, I posit an example of reading the code by way of tracing out use cases. Admittedly, tracing out processes from start to finish is only one of an infinite array of reading practices regarding code. Most notably, many, including the developers, encounter code through an integrated development environment (ide), which allows them to traverse the code with much greater alacrity, leaping from function to definition, moving through multiple files fluidly, et cetera. What I am narrating here is the process by which someone might trace out the functioning of the code from start to finish of the software, again, a kind of walking through the code.

[15] GPX is the name for the XML GPS data schema that the Tool uses.

[16] Unless otherwise noted, the lines of code are taken from the TBMIDlet.Java file. The line numbers used were added to this file as JAVA does not require line numbers.

[17] Though this is the constructor, the application does not launch until the start method is called.

[18] Line 48 of DowsingCompass.Java reads

```
TBGpxParser gpxParser = new TBGpxParser(TBGpxParser.RES, "../..../.." + gpxFileName);
```

This line essentially prevents the use of a file from a URL by pushing the file structure back (../) through a series of containing folders.

[19] See <http://scalar.usc.edu/nehvectors/border-codes/water-witching>.

Works Cited

- Bakhtin 1982** Bakhtin, M.M. *The Dialogic Imagination: Four Essays*. Austin: University of Texas Press, 1982.
- Borovoy 2001** Borovoy, Rick, et al. "Folk Computing". Presented at *SIGCHI Conference on Human Factors in Computing Systems* (2001). <http://dl.acm.org/citation.cfm?id=365316>.
- Camnitzer 2007** Camnitzer, Luis. *Conceptualism in Latin American Art: Didactics of Liberation*. Austin: University of Texas Press, 2007.
- Cantwell 1993** Cantwell, Robert. *Ethomimesis*. Chapel Hill: University of North Carolina at Chapel Hill Press, 1993.
- Carroll 2011** Carroll, Amy Sara. Transborder Poetry. 4 Jan. 2011. E-mail.
- Carroll 2012** Carroll, Amy Sara. 12 Mar. 2012. Skype.
- Dislocative 2010b** Dislocative Meida. *Dislocative Media: Transborder Immigration Tool as Aesthetic Sustenance* [7/11]. 2010.
- Dislocative Media 2010a** Dislocative Meida. *Dislocative Media: Transborder Immigration Tool as Aesthetic Sustenance* [5/11]. 2010.
- Douglass 2010** Douglass, Jeremy. "Comments on Comments in Code". Presented at *Critical Code Studies*, sponsored by University of Southern California (2010). <http://thoughtmesh.net/publish/369.php>.
- Dunbar 2009** Dunbar, Alex. "FOLLOW THE GPS, ÉSE - The Transborder Immigrant Tool Helps Mexicans Cross Over Safely". *Vice* (November 2 2009).
- Dworkin 1993** Dworkin, Craig. *Introduction. Anthology of Conceptual Writing*. 1993. <http://www.ubu.com/concept/>.
- Hunter 2009** Hunter, Rik. *The Literacy of Proceduracy: A Conversation with Annette Vee*. HASTAC. September 18 2009. <http://www.hastac.org/blogs/rikhunter/literacy-proceduracy-conversation-annette-vee>.
- Marino 2006** Marino, Mark. "Critical Code Studies". *Electronic Book Review* 4 (December 4 2006). <http://www.electronicbookreview.com/thread/electropoetics/codology/>.
- Miller 2010** Miller, Joshua Rhett. "Critics Blast Transborder Immigrant Tool as Irresponsible Use of Technology". *FoxNews.com* (March 10 2010).
- Morlan 2010** Morlan, Kinsee. "After the Storm". *San Diego Citybeat* (September 8 2010).
- Najarro 2007** Najarro, Jason, Brett Stalbaum, Micha Cárdenas and Richard Dominguez. *The Transborder Immigrant Project*. 2007. <http://bang.calit2.net/>.
- Perens 1999** Perens, Bruce. *Open Sources: Voices from the Open Source Revolution*. Newton: O'Reilly Media, 1999.
- Raley 2009** Raley, Rita. *Tactical Media*. Minneapolis: University of Minnesota Press, 2009.
- Stalbaum nd** Marino, Mark. *Brett Stalbaum Complete Interview : Mark Marino*. Internet Archive. <https://archive.org/details/BrettStalbaumCompletelInterview>.
- The Electronic Disturbance Theater 2011** The Electronic Disturbance Theater. *The Electronic Disturbance Theater at USC*. University of Southern California: 2011.
- Thompson 2011** Tok Thompson. 09 2011. Personal Interview.
- Urrea 2005** Urrea, Luis Alberto. *The Devil's Highway: A True Story*. New York: Back Bay Books, 2005.
- diSessa 2011** diSessa, Andrea. *Changing Minds: Computers, Learning, and Literacy*. Cambridge: MIT Press, 2001.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

