

Channels of Ventriloquism in New Media Artworks

Victoria Bradbury

CRUMB: Curatorial Resource for Upstart
Media Bliss, University of Sunderland,
Sunderland, United Kingdom

vebradbury@gmail.com

ABSTRACT

This paper extends previous research relating puppetry to new media art. Two of the author's artworks, *Toast* and *Ventriloquisms for Fun and Profit*, are used as case studies. The projects are summarised and insights that emerged from their implementation are proposed as results. Throughout the text, ventriloquism diagrams are used to illustrate possibilities for directional transmission of speech and to reflect upon the roles of code, artist, and participant in new media artworks.

KEYWORDS

Code; Speech; Ventriloquism; Participation;
Translation; Voice; Performativity.

1 | INTRODUCTION

DUM: Hey, what's goin' on here? What's the idea of wearing the white coat?

VENT: Well, you see, you are the patient, and I am the dentist.

DUM: (Calmly) Oh, I see . . . (Suddenly leaps up) WHATTTT? (Winchell, 1954)

Computer code is written, compiles, and then runs through an end user. In the act of ventriloquism, the voice is thrown to appear to be coming from somewhere other than the original source. A study of ventriloquism in relation to new media art considers the origins of voice and code, the phenomenon of one



Figure 1 | Zelda Zilch with Burke Bradbury, Lower Providence, PA, USA, 1992.

entity speaking through another, and potentialities of control in computational systems.

Oscar and Zelda Zilch are the dummies in my family; they are 5 foot tall and made of wood. My late grandfather, Burke Bradbury, was an amateur ventriloquist (Figures 1 and 2). I grew up with his acts, including call and response songs, a cover of Abbot and Costello's "Who's on First?" and, my favourite, when the dummy's mouth joint "breaks" and requires repair with a ridiculously long screwdriver. My grandfather takes Oscar's head apart in two halves, twists screws and tweak springs, while the two banter and Oscar begs for mercy.

While ventriloquism and puppetry have been related to code by various authors within new media art discourse, a thorough examination of the relationship between code and ventriloquism has not been made. This paper begins by drawing together key references. Next, a ventriloquism analogy is used to consider the roles of code, artist, and participant in two of my new media artworks, *Toast*, which explores translation and *Ventriloquisms for Fun and Profit*, a performance with call and response song. Ventriloquism diagrams are used as a tool to analyse channels of vocal transmission that occur in artworks discussed. An initial model for a code-based participatory artwork looks like the diagram in Figure 3.



Figure 2 | Oscar Zilch with Burke Bradbury, Bradbury Family Reunion talent show, Ocean Park Maine, USA, 2003.

The ventriloquist is responsible for both sides of a conversation. While the dummy offers the illusion of autonomy, the audience knows that he is an extension of the ventriloquist (St. Clair, 2008). Thus, a doubling occurs and a loop is established:

{Ventriloquist Voice → Dummy Voice → Ventriloquist Voice →}
 == (is equivalent to)
 {Ventriloquist Voice → Ventriloquist Voice → Ventriloquist Voice →}

In the essay, *Interaction/Participation: Disembodied Performance in New Media Art*, Beryl Graham states:

Conversation is a highly elaborate skill involving exchange, evolution, creativity, interpretation, empathy and ambiguous language. Computer logic may just about be able to manage the first two factors, but

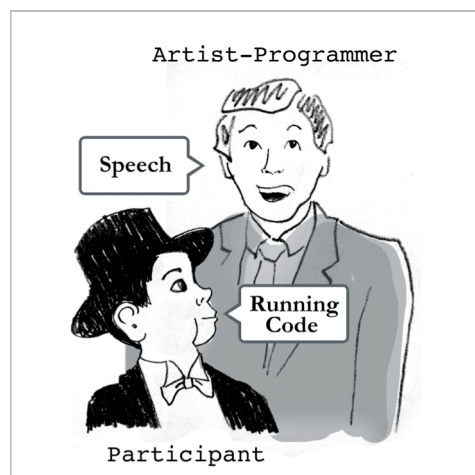


Figure 3 | Initial ventriloquism diagram for code-based participatory artworks.

beyond that it needs firm rules and predictable structures. (Graham, 2007b)

Languages, both coded and spoken, provide the ability to communicate. Whilst coding languages act, as Graham states, they are not capable of carrying out complex conversation. Paradoxically, in our increasingly technology-dependent culture, code is the intermediary between many conversations. In this act of mediation, code limits and directs the way in which conversations unfold.

Just as an audience watching a ventriloquism performance ignores the objecthood of the dummy in order to be entertained, users similarly ignore the coded infrastructure beneath computational devices. On tablets, for example, apps limit users to only behaviours made available by the programmer, while those using the technology are locked out of adjusting it or accessing its inner-workings [1].

New media artworks can highlight systems of control structured by code. During the March 2014 CRUMB discussion on the topic of *The Performativity of Code* [2], new media artist Jack Stenner (Stenner, 2014) wrote that what interests him in his current practice “are various strategies to unpack ideology in code, or ways power, though code, is formed/masked via its operation in culture.” This paper analyses these power structures through an analogy of ventriloquism.

2 | VENTRILOQUISM AND NEW MEDIA ART

When an artist writes lines of code for a work that invites visitor participation, the code (the voice of the artist) is speaking (is thrown) through the visitor’s actions. But what role does this visitor take when interacting with the code? Is he simply a dummy, acting as a medium for the coder’s voice?

Curt Cloninger (2010) states that “computers don’t execute code in a transcendent, metaphysical vacuum...Code is run on physical hardware in lived and present (albeit massively accelerated) time.” In order to come into being, code “has to be read by and run on something—a person or a computer.” During this performative moment, the code is united with both the hardware on which it runs and with the person who interprets the result of this running. If this is true, then each time a program runs, a unique organism emerges.

Brian Massumi (2002) considers code to be strictly protocol while bodies are analog and continuous. Thus, code can “potentialize, but only indirectly, through the experiential relays the reception of its outcomes sets in motion ... Whatever inventiveness comes about, it is a result not of the coding itself but of its detour into the analog.” This means that the body is that which translates the strictly pre-determined code into the analog.

Beryl Graham (2007a), in her essay *The Dark Side of Light: Ambivalent Interaction*, uses the term “reverse puppetry” to discuss Lozano-Hemmer’s *Body Movies*. Graham states that *Body Movies* “works on at least three interactional levels”: the audience covering projected people with their bodies, collaborating to cover the images, at which point the program produces more images. The third level is when the audience “has full creative input to play with the shadows—running the full gamut of human power relationships—flirting, mock sex and violence, or more complex narrative improvisation.” In *Body Movies*, the audience takes the role of the puppets. While a puppeteer is normally manipulating shadow puppets, Lozano-Hemmer turns this relationship on its head, giving the “puppets” the creative autonomy to control the show.

Ken Feingold’s artworks directly involve ventriloquist dummies or disembodied heads. In Feingold’s *If/Then*, a conversation between two mechanical, programmed heads is generated in real-time with language-processing software. Mark Hansen discusses *If/Then*, stating that

Feingold’s work compels the viewer-participant to wonder whether her own speech is not itself perhaps the equivalent of the computer’s anguished and often confused vocalizations, and what this potential equivalence says about the process of communication itself. (Hansen, 2004)

Jonah Brucker-Cohen’s artwork merges code and object with an element of visitor participation. In *Alerting Infrastructure*, a visitor to a website affects the behaviour of a series of objects. Figure 4 shows a website hit counter that slowly destroys a building.

When asked in an email interview about the word “ventriloquism” in relation to his practice, Jonah



Figure 4 | *Alerting Infrastructure!* Jonah Brucker-Cohen (2003) Installation at Social(DIS)order show at LSU.

Brucker-Cohen stated that it is relevant “if you are making work that allows the user to be heard through some other object – i.e. not themselves.” (J. Brucker-Cohen, personal communication, June 08, 2012) In this description, the participant’s voice, rather than the artist’s, is being thrown through an object (Figure 5).

The ability or privilege to speak grants power because the voice can be used to direct others to take certain actions, to persuade, or to assert oneself as an individual in the world. Geoff Cox and Alex McLean’s book *Speaking Code* begins with a quote by Theodor W. Adorno from “Institute for Deaf-Mutes” that contextualises ventriloquism within social power structures:

“While the schools drill human beings in speech ..., the pupils become increasingly mute ... In the all-embracing system conversation becomes ventriloquism.” (Cox & McLean, 2013)

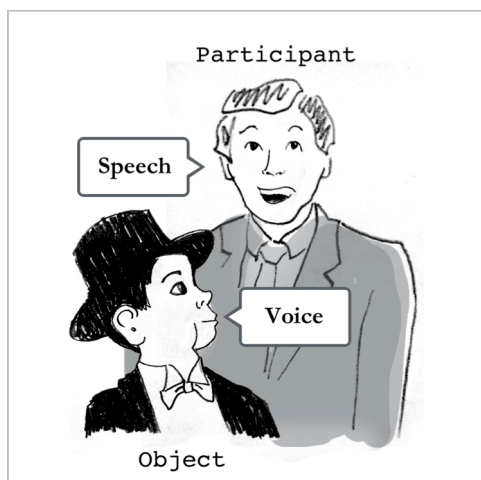


Figure 5 | Ventriloquism diagram with participant and object.

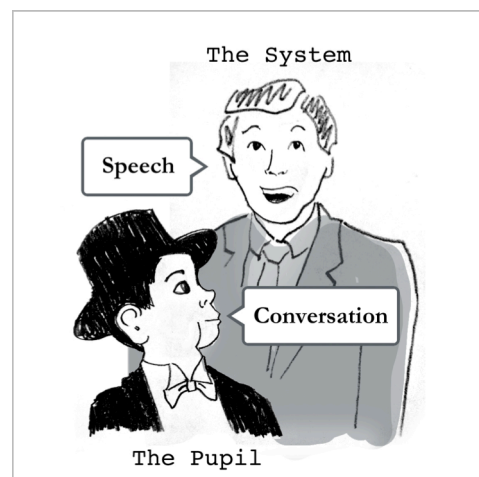


Figure 6 | Ventriloquism diagram for system-pupil relationship.

In Figure 6, the pupils mouths speak someone else’s words. Like a ventriloquist’s dummy, the pupils do not have the autonomy to articulate their own thoughts with their bodies and voices.

Figures 3, 5 and 6 illustrate directional transmissions of speech. Figure 3 shows an artist-programmer as the ventriloquist and the code running through the participant as a dummy. In Figure 5, the participant is the ventriloquist whose speech moves through an object. Figure 6 illustrates systems of power and social control as the system as ventriloquist dictates the speech of pupils.

A ventriloquism analogy has been considered in two ways. First, the dummy as a powerless object who simply channels the voice of the ventriloquist through his mouth. Second, the dummy as a double of the ventriloquist, the same voice appearing to emerge from another body. These models represent participant and artist roles in new media artworks, opening for reflection whether a participant has autonomy to act when they engage with artist-written code.

The following case study of *Toast* considers the projects’ use of code as a medium to translate and display the voice of participants. Channels of ventriloquism that are present in the work are also analysed.

3 | TOAST

Toast uses a coded translation device to mediate the speech of a performing participant (see Figure 7). The project was initiated in 2011 while I was living in China with Mandarin language ability that limited me to only simple utterances. Although I could speak enough to purchase food at a local market, to relay basic directions to a taxi driver, or to tell someone my occupation and nationality, the attempts at discussion that ensued after these basic exchanges discouraged further conversation. In Beijing, I quickly became interested in making a translation device that would allow me to take conversation to a more complex level while emphasising the ridiculousness of using a machine to communicate rather than taking the proper steps to learn a language.

Work on *Toast* began by moving directly into the code using translation in a Processing sketch [3]. I drew upon libraries to handle the speech-to-text functionality and the Google Translate integration [4]. Next, I began to search for agents of performativity that were already present in Chinese culture that could help participants to overcome potential shyness when speaking into the device.

During the initial stages of project research, I was attending various functions in and around Beijing and Shanghai including gallery openings, private dinners, and banquets. It occurred to me during these occasions that there was something special going on in the performance of a toast. In a landscape where public expression is not widely encouraged, the toast provided a forum for a person to express his views and emotions about the occasion at hand and his gratitude to guests or hosts [5]. I decided to draw



Figure 7 | Feijiacun, Beijing iteration of *Toast*, Open Studio Exhibition curated by Filipa Reis (2012).

upon the social code of the toast in my emerging project, as it was a performative gesture with cultural precedent.

Iteration, a property inherent to media artworks because of code's flexibility, was important to *Toast* because the project resulted in a series of tests rather than a single work. These tests included an audience-performer format at Barcamp Shanghai, a series of one-on-one experiments at Shanghai's Xinchejian Makerspace, and an installation prototype at the *Feijiacun Shangri-La Art Community Open Studio Exhibition* on December 1, 2012. This final iteration is described below.

The Feijiacun *Toast* installation included instructions for visitors that were posted on the wall in both English and Mandarin. The text asked a participant to address his toast to an adjacent photograph of a common restaurant table, set with empty chairs in the round. This table image served as a blank canvas on which the visitor could imagine people seated for a meal.

The participant approached the computer, read the instructions, picked up the microphone in one hand (and optionally an empty wine glass in the other), and then spoke a toast to his imagined companions at the dining table. His words were sent through the Processing sketch. Here, speech was turned into text in the spoken language. This text was sent to Google Translate where it was translated it into the "opposite" language (English <> Mandarin), and then was sent back to be displayed on the screen. Throughout the interaction, a webcam picked up a live-feed of the speaker's face, which was situated next to a speech bubble containing the final result of the translation.

In *Toast*, the translations returned by the code were almost always inaccurate and not a representation of what the speaker had actually said. This defeated the initial purpose of the project: to help a non-native speaker to be better understood. Instead, it highlighted the ineffectiveness and potential humour of machine translation.

The excerpt of code in Figure 8 is activated if there is a button-press by a user, at which time the code "hears" the spoken language, turns it into a string, sends the string to be translated, then returns the result to be displayed.

In Figure 8, the variable *result* initially represents *utterance*, or the words spoken by the participant. Inside of the *if* statement, *result* becomes equivalent to the translation (*translatedText*). While the translation algorithm considers the original utterance and the result of the translation to be equivalent, the human participant knows that the final translation is often quite distant from what was actually said.

Although the code imposed translations on the *Toast* participants, they were free to interpret the text and image, drawing additional meaning or humour from the juxtapositions provided. Because most of the people attending the Feijiacun exhibition could speak some English and some Mandarin, among other languages, the participants were aware of these missed translations. This understanding led to them becoming actively engaged with the piece, gathering in groups, and creating a playful performative atmosphere around the spoken utterances, the doubling of a participant's likeness on-screen, and the floating speech-bubble translations.

A ventriloquism diagram for *Toast* looks like the diagram in Figure 9, where the ventriloquist embodies the translation. This translation is dictated by an algorithm, which sends speech back through the mouth of the participant.

4 | VENTRILOQUISMS FOR FUN AND PROFIT

In *Ventriloquisms for Fun and Profit*, I took the role of coder and performer as an audience was invited to participate through call and response song (see Figure 10). In this performance, the dummy was a puppet of a cat, coded in openFrameworks rather than built with wood, strings, and glue.

The performance took place on 26 April 2013 at Datarama, Pixel Palace, Newcastle, UK. The piece began by engaging the audience in a song by

```
println(utterance);
result = utterance;
if (buttonCaller == 1){
String
    translatedText=Translate.DEFAULT.execute
    (result, Language.ENGLISH,
    Language.CHINESE_SIMPLIFIED);
println(translatedText);
result = translatedText;
}
```

Figure 8 | code snippet from *Toast* processing sketch.

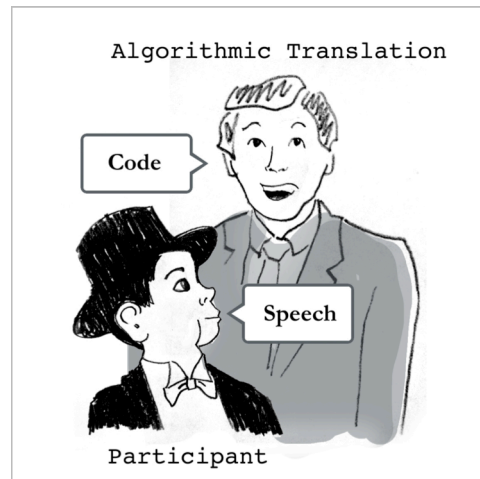


Figure 9 | Ventriloquism diagram for *Toast*.

instructing them to repeat the phrase “Oh Mona” after each artist-led line of verse. Between verses, they sang along with a chorus, “Oh Mona you shall be free...”. The text to be sung was displayed onscreen. When written in pseudo code, these instructions to the audience create an if-else-statement:

```
if (line of verse is complete){
    Sing "Oh Mona";
};

else if(entire verse is complete){
    Sing chorus;
};
```

At Datarama, the audience willingly participated, singing along and “joining in” or following the instructions. When everyone in the room was singing, social codes enforced individual participation.

To begin the performance, I changed a Boolean value in the code from false to true in order to get the dummy “working” (See Figure 11 below). This moment of live coding referenced my grandfather’s

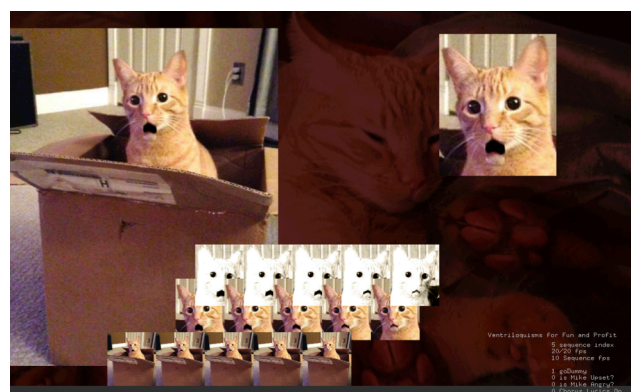


Figure 10 | *Ventriloquisms for Fun and Profit* screenshot (2013). Audio asset: <http://blurringartandlife.com/vb/ventriloquisms.html>.

act in which the dummy’s head is taken apart in order to get it functioning again.

When I changed the value of `makeDummy` from `false` to `true`, the dummy appeared to have suddenly gained the ability to move his mouth. The code-saavy Datarama audience laughed at this moment (see Figure 11).

After the song, I performed a ventriloquist act, *At the Dentist*, from *Ventriloquism for Fun and Profit* (Winchell 1954) the same book used by my grandfather as an instruction manual for constructing his dummies.

Winchell’s *At the Dentist* sketch follows a common trope of ventriloquist performances that create humour through violence between the ventriloquist and the dummy. At the beginning of *At the Dentist*, the dummy discovers that he is a patient at a dentist office and the ventriloquist is the dentist. The humour here lies in the fact that the dummy is surprised to find out that he is going to be subjected to a potentially painful procedure. There is an underlying power structure between a dentist and a patient similar to the pupil/system dichotomy seen in Figure 6. This role of the dentist as an outside expert on a physical problem is also reflective of power structures inherent to the field of medicine in which doctors are considered experts who wield power over knowledge of how to heal or “fix” the human (especially female) body. Data is similarly considered an expert, a qualifier, and a way to log information about whether or not an individual is sick, well, a criminal, a terrorist, a good guy, or a bad guy. In relationships of unequal power, the dummy, the patient and the pupil are at the bottom rung, subject to the speech and authority of the ventriloquist, the dentist, or the teacher. A *Ventriloquisms for Fun and Profit* diagram might look like Figure 12.

In Figure 12, the ventriloquist is equivalent to the artist-programmer while the dummy represents the audience as they sing the chorus of “Oh Mona”. The artist authored the code that creates the dummy and her voice performs the scripted ventriloquist act and leads the call and response song.

```
//makeDummy
makeDummy = true;
```

Figure 11 | “makeDummy” code.

5 | REFLECTION AND CONCLUSION

In *Toast*, while participants speak through a translation device that doesn’t translate accurately, the code becomes the ventriloquist, imposing meaning on the speaker’s image. The code places words in the mouth of the participant who is left to interpret the translation as it hovers beside his face-image (a doubling of the self). While code is the ventriloquist, the participant is the dummy with a sense of interpretive autonomy.

Ventriloquisms for Fun and Profit situates the artist and audience as performers within a system dictated by artist-written code in which underlying social codes influence audience participation. While the audience is the dummy in Figure 12, the artist is also a dummy during the performance, as both parties are controlled by the code and the code (as ventriloquist) speaks through them.

In a conversation, words are spoken by one party, then heard, considered, and responded to by another. This exchange continues in a loop. In a toast, one person speaks to a group in a performative moment. An audience hears this speech and clicks their glasses, initiating a consecration of the words. In a call and response song, one person holding the power of performance sings a line and a group responds with a pre-established, repetitive phrase. In ventriloquism, the ventriloquist speaks, but simultaneously and in another voice, channels his speech through the dummy.

Jack Stenner (2014) has written that humans are “the ‘neuronal’ support for technology.” This view reflects the same union of code and body as the analogies of

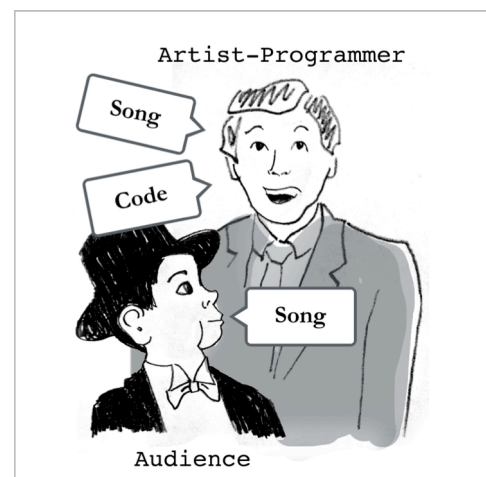


Figure 12 | *Ventriloquisms for Fun and Profit* diagram.

ventriloquism described in the above case studies. In viewing a ventriloquist performance and while interacting with code, an audience or participant accepts and ignores the workings behind the scenes in order to accept the illusion. In ventriloquism, the trick is obvious, but with code, layers of obfuscation, translation, and compilation hide the source, making it unclear exactly how the program controls the participant's actions.

In each of the analogies of ventriloquism explored here, code is equivalent to the author and participant as it runs through all of the entities within the system. Bodies and voices are not separate from code.

ENDNOTES

[1] The experience of technology, for the majority of people, does not include the creative act of writing code, but only the consumption of an interface with no entry into its inner workings.

[2] CRUMB (<http://crumbweb.org/>) run a New Media Curating discussion list that proposes month-long discussion topics with list members and invited participants. The March 2014 topic was *The Performativity of Code* and was mediated by CRUMB researchers Victoria Bradbury and Suzy O'Hara with 17 invited respondents. The full discussion may be found on the CRUMB online archives.

[3] Processing is a Java-based programming language created at MIT Media Lab by Casey Raes and Benjamin Fry primarily used by artists and designers to create animations, generative images, or interactive artworks.

[4] Florian Schulz's 2011 STT Library was used for speech to text (<http://www.getflourish.com>) and for Google Translate integration, the Google Translate API (<https://developers.google.com/translate/>)

[5] In Britain, a prominent part of a wedding ceremony is the series of toasts traditionally made by men in the wedding party (the best man, groom, and father of the bride). In both Chinese and British culture, the toast is situated as an official forum in which expressing emotion and sentiment is made socially acceptable by the formality of the performative act.

REFERENCES

Cloninger, C. (2010). GlitchLinguistx: The Machine in the Ghost / Static Trapped in Mouths. Lab404.com. Retrieved 6 November 2011, from <http://lab404.com/glitch/>

Cox, G., & McLean, A. (2013). Speaking code (1st ed.). Cambridge, Mass.: The MIT Press.

Graham, B. (2007a). The Dark Side of Light: Ambivalent Interaction. In R. Lozano-Hemmer & R. Hill David, Under Scan (1st ed.). Nottingham: East Midlands Development Agency.

---. (2007b). Interaction/Participation: Disembodied performance in New Media Art. In J. Harris, Dead History, Live Art? Spectacle, Subjectivity and Subversion in Visual Culture since the 1960s. (1st ed.). Tate Liverpool Critical Forum, Volume 9. Liverpool: Liverpool University Press & Tate Liverpool.

Hansen, M. (2004). New philosophy for new media (1st ed.). Cambridge, Mass.: MIT Press.

Massumi, B. (2002). Parables for the virtual (1st ed.). Durham, NC: Duke University Press.

St. Clair, J. M. (2008). Novel Listening: Background Sound and the Technologies of Aural Control. Dissertation Abstracts International, Section A: The Humanities and Social Sciences, 68(9), 3859.

Stenner, J. (05 March 2014). "Re: [NEW-MEDIA-CURATING] March Discussion Begins: The Performativity of Code." New Media-Curating Discussion List. Available from URL: <<http://www.jiscmail.ac.uk/lists/new-media-curating.html>>.

---. (17 March 2014). "Re: [NEW-MEDIA-CURATING] [NEW-MEDIA-CURATING] Can non-human entities perform?." New Media-Curating Discussion List. Available from URL: <<http://www.jiscmail.ac.uk/lists/new-media-curating.html>>.

Winchell, P. (1954). Ventriloquism for fun and profit (1st ed.). Baltimore: I. & M. Ottenheimer.

BIOGRAPHICAL INFORMATION

Victoria Bradbury is a visual artist weaving programming code, physical computing, body and object. She is based in Newcastle UK and studies the performativity of code as a researcher with CRUMB. Her work has been shown at the Albright Knox and Hallwalls Galleries (Buffalo), Harvestworks and Cuchifritos (NYC), and Globe Gallery (Newcastle). She

participated in the Shanghai Biennial IMMERSION: Art and Technology workshops (2012) and was a resident artist at Digital Media Labs in Barrow UK (2014) and Imagine Gallery, Beijing (2008). Bradbury has presented research at xCoAx (Porto), ISEA 2013 (Sydney) and Re:Wire (Liverpool, 2011). She is a collaborator with Attaya Projects and a member of the New Media Caucus board.
<http://www.victoriabradbury.com>