Useless Utilities

Saul Albert

www.twenteenthcentury.com/saul

"The useless alone is truly beautiful; everything else is ugly, since it is the expression of a need, and man's needs are, like his pitiful, infirm nature, ignoble and disgusting. - The most useful place in the house is the latrines"

Théophile Gautier's 1835 Preface to *Mademoiselle de Maupin* is often cited as the "manifesto" of the romantic notion of "Art for art's sake". While his feelings about his fellow humans seem thankfully outdated, it is surprising, and disturbing that his feelings about aesthetics are still widespread, and even worse, misunderstood.

There is a common misconception in art and technology crossovers that any cultural product can become art if it is robbed of it's utility, that the product of a scientific or technological process is art if it has been done simply because it can be done. This misunderstanding replaces "art for art's sake" with "anything for it's own sake".

This is the logic behind the pretty microscope photographs of dyed cells that pharmaceutical companies are so fond of hanging in galleries, the aesthetically pleasing by-products of their "too-complicated to explain to the public" experiments. "Art for public relation's sake".

In the context of tech-art culture this logic has produced an even more horrible misunderstanding: "art for technology's sake". The reliance on some kind of unconscious, artistic intuition performed in front of a computer has resulted in the installation of countless adverts for Macromedia, Apple, Sony and other culturpreneurial technology companies in high-profile art galleries around the world.

The problem for artists who do not want to be unpaid advertising executives is that without careful and critical attention to the processes and imperatives of software, their work can be processed into bland "content" and aesthetic pleasantry through an unacknowledged collaboration with corporate software.

Utility is also the myth of the software tool. Personal experience as well as statistics indicate that computer use can slow productivity and create huge expense and inefficiency.

Technology companies cover over this absurdity by creating and fostering needs for their software's unnecessary and distracting "features", creating problems by anticipating them with software solutions. Compatibility problems are built into software to leverage increased market share and even worse "mind share".

This term is used constantly by software marketing people, but never defined. The reason for this becomes clear when we try. Market share can be defined as "a company's control over consumer spending expressed as a percentage of the sales for the total industry". So with this in mind, the term "mind-share" takes on a sinister, Orwellian meaning.

The ultimate goal of the company is to become the "industry standard" (100% market and mind share), entirely framing the work of the media producer, writer, or artist with the rules and potentials of their software. And by framing the work of the media producer, their intentions filter down to the media consumers.

This situation can be seen in terms of a hierarchical food chain diagram, with the media consumers, the public at the bottom, their mind-share being consumed by the media producer, whose aggregated share is in turn consumed by the programmers of their software tools.

```
       software producers
             -----
           /|01*10|\
          | |0***0| |
           \|01*10|/       +------------+
             --+--         |intervention|
               X           |point 2     |
              / \<---------+------------+
             /   \
        media producers
           |       |
           v       v
       0  +-+    0  +-+
       +\-| |    +\-| |
       | *+-+    | *+-+
       +-+--+    +-+--+    +------------+
            / | \          |intervention|
           /  |  \         |point 1     |
          /   |   \<-------+------------+
         /    |    \
       media consumers
       /      |      \
      /       |       \
     /        |        \
     |        |         |
     v        v         v
+-------+ +-------+ +-------+
|  _O_  | |  _O_  | |  _O_  |
|   |   | |   |   | |   |   |
|  / \  | |  / \  | |  / \  |
+-------+ +-------+ +-------+  figure 1
```

Computer technologies of media viewing or "browsing", and particularly the web browser capitalise on mind-share by selling a percentage of their mind-share to the highest bidder, and then framing the act of looking, of absorbing information with advertising.

Artistic software projects have often intervened at this point in the hierarchy, between the media producer and the consumer (see fig. 1 - intervention point 1).

The classic example is I/O/D's Webstalker, a lean, stripped down browser that cuts through the distracting visual complexity and commercial glare of the web and reveals the quietly expanding information framework underlying it. This and the minimalist, unfamiliar interface breaks the integrity of the simulated "Desktop", the graphical user interface and it's extension to the web through the browser.

The 1980's video game style graphics of Nullpointer's Webtracer software has a similar aim and method, to unsettle the metaphor of browsing and our casual acceptance of that single visualisation of the web as a 2 dimensional shopping mall.

Jodi's "%wrong browser" collection is the least "user-friendly" of these "art browsers." The usual situation in which the browser is produced by a single multinational corporation and then used globally is turned on it's head. Each of the %wrong browsers has a domain name. ".co.kr (Korean) ,.nl and the ubiquitous .com and .org. Each of these browsers exhibit a difficult, illegible interface that slowly reveals a kind of personality for each browser. The Korean browser is very nervous, has no words or symbols, only a shivering cursor on a dark screen that suddenly erupts with scrolling colour and text like a malfunctioning Times Square billboard display. ".com" is flamboyant, brightly coloured and social, graphically linking roughly drawn browser windows and happily spreading the html thickly over the screen. ".org" is a monstrous bureaucrat-printer, maniacally documenting and recording every site that it visits in semi-sensible ASCII and saving it's mutilated source code as text files all over the computer. ".nl" is a reserved, dry gatherer of information with a low-tech green-screen aesthetic. The behaviours of each browser could be seen to correspond to putative national and multinational domain name identities and stereotypes. Each flattening out the user's experience of the disparate spaces of the web into a continuation of their own fractured personalities, just as one might argue that Internet Explorer or Netscape Navigator flatten out the disparate spaces of the web into a corporate mono-culture.

Over the last few years there have been many more "art browsers" (i) , and although these projects make very astute comments about the state of the web and the browser, the potential of software as an artistic medium offers much more than this kind of software-art "intervention".

In "A means of mutation", Matthew Fuller describes the definition which the Webstalker aims to fulfil as "not just art". A piece that would be relevant in multiple contexts, could move between use value and conceptual value seamlessly. "Not just art" rejects the dead end-dichotomy of culture vs. counter-culture and suggests hybridised, developmental, unstable cultural forms that can sustain themselves outside of art's frame of reference and financial backing.

Although there are ways to make use of the Webstalker in a non-art context (visualising the structure of web sites for development purposes), no development team has emerged with new ideas for how to improve it as a tool, Webstalker 2 was promised, but never emerged. "Not just art" did not happen with the

Webstalker, and in "means of mutation" Fuller almost acknowledges this. He calls it "tactical software" and observes that it's development was limited by money, time and available skills. As tactical software the Webstalker was very successful, generating huge amounts of media attention, critical thinking and inspiring further developments in art and software, but I would argue that to become "not just art" the artwork must have utility outside of the frame of art.(ii)

In the diagram above, the most influential position is clearly that of the software programmer, and the most obvious point for intervention is there, between the software producer and the media producer. (See fig.1 intervention point 2).

Two artistic software projects that fit into the category of media production tool, and function as "not just art" are Auto-illustrator by Adrian Ward and b1257+12 by Netochka Nezavanova.

On first inspection Auto illustrator looks like a standard vector graphics program. However, once you start using it, the quirky interface and the difficulty of making the program behave as expected makes explicit the fact that another agenda is at work.

A favourite features are the bugs; rule-based automata that drag lines of colour around behind them. This is a playful reference to the 60's Fluxus art practice of dipping insects in ink and letting them draw a path over paper as a way of challenging conventional notions of authorship. In this case it becomes a light-hearted joke and an incitement for the user to engage with Ward in a struggle over the authorship of the piece. Certain vectors of control are available to the user (add and remove bugs) but the behaviour of the bugs is determined by the algorithms Ward used to program them. In this way a third party is brought into the struggle for the authorship of the piece: the generative code that underlies many of the features of Auto-illustrator.

I am no expert, but for the purposes of examining this artwork a quick introduction to these ideas may be necessary:

Biologist Aristid Lindenmayer gives his name to the mathematical modelling of growth patterns in nature, trees, leaf structures etc. The generative grammars developed to model these processes work by recursive development of a limited set of symbols. For example, if we start with a phrase: AGGDB, and say that for every iteration we replace A with AGG, G with DDA, B produces B and D produces BBA.

A=AGG
G=DDA
D=BBA
B=B

on the first iteration we have

AGGDB

on the second,

AGGDDADDABBAB

on the third

AGGDDADDABBABBAAGGBBABBAAGGBBAGGB

past here it is better done by computer.

If A meant "draw a line 10 pixels long" and G meant "move left 10 pixels" you can see how this kind of generative code can produce unpredictable but formally coherent visual developments.(The coherence is due to the inevitable self-similarity of the designs).

Emergent behaviour systems work in a similar way, a simple set of rules is given to a bug, for example : walk forward, turn a random number of degrees every 20 paces and don't bump into other bugs or drawings. When you have one bug, a human can anticipate the results of this behaviour quite well. Once there are 200 bugs, all avoiding each other, and changing each others paths, the complexity is immense and again, the results cannot be anticipated without computer modelling.

So by incorporating these kinds of algorithms into Auto-illustrator there is a loss of authorial control by all parties, neither Adrian Ward, nor the user, nor the computer can completely determine the outcome of their collaboration.

This complex conception of authorship as a kind of running battle has been extended in later releases by the "Swap Artwork" plugin. While working on a drawing, the user can apply the "Swap Artwork" filter which uploads the user's image to the auto-illustrator server and swaps it for an image being worked on by another Auto-illustrator user. Yet another player is brought into the authorship competition: the collective user group of the program at any one time.

The effect of all these generative features and sudden, worrying distortions of the artwork is alternately fascinating and aesthetically horrible.

Netochka Nezavanova's "b1257+12" which she describes as a real-time interactive sound processor is similarly difficult to manage.

At first the interface is completely illegible, lists of numbers respond to mouse or keyboard activity, sliding scales with no labels respond anti-intuitively to mouse movements. Even if importing and working on a familiar sound file, it is unclear whether the tumult of sound emerging from the machine are being effected by the user's activity at all.

However, after playing and experimenting for a long time it is possible to tease a method of use out of the software, find ways of behaving and moving that for some reason produce a desired sound. In this way, using b1257+12 becomes a very personal experience, each user determines their own technique while at the same time, random re-configurations of key commands and responses constantly alter the programs functioning, undermining this familiarity with the software, forcing the user to start again.

What both these artworks succeed in doing is making explicit the hidden struggles and difficulties of conventional software. For example, the struggle for authorial control with Photoshop is less visible. Photoshop hides that struggle in a hugely complex, slick interface where the designer is offered a million options in a million pull down menus to give the illusion that they are making choices and are in total control of what they are doing. In both Auto-illustrator and b1257+12, the subjective presence of the author is always felt. The user is never allowed the comforting illusion of control.

As Adrian Ward says "When someone uses my software, it's me!". Artistic subjectivity, so often hidden by the dry, fleshless aesthetics of computer based art, is a vital and visible part of both these projects. The humorous and dysfunctional human-computer interfaces become interfaces between the viewer and the artist. The software takes on the programmers artistic subjectivity and engages the user in dialogue, organising and interrupting their process and final product. It is this process of negotiation and compromise between the artist-programmer and the media producer that makes the product interesting.

In both cases, the persona of the artist/programmer is pushed beyond the limits of the software. Netochka Nezavanova (or one of her many selves as integer, antiorp or m9ndfuck) is notorious for asserting her persona into mailing lists. She writes copiously, posting provocative, sometimes callous, sometimes poetic texts to many lists, often creating mayhem and discord by appealing to some and antagonising other members of the list, dividing them on the issue of whether she should be banned or not. The content of her posts is usually infused with belligerent views about authorship and intellectual property (she will often claim authorship and threaten to sue people she considers to have stolen her ideas), or she'll make back-stabbing personal tirades against people in the net art scene who she has taken a dislike to. In other contexts (the support mailing list for one of her software tools for example) she is very helpful and always responds to intelligent queries within a few hours. A friend of mine who has purchased her most popular software " Nato+55" and is on the mailing list tells me that whenever he emails her he is in constant fear that the question might have been answered already in the support forum, and he'll get flamed.

The texture of her email exchanges also seems to have been algorithmically processed, re-purposing old BBS traditions such as ASCII art and "hack-speak" certain roman characters are replaced with unpronounceable punctuation marks or numbers. This language-game (which she calls KROP3ROM||A9FF) mixes and remixes Russian, English, French, and German vocabulary in the same post which may contain varied cultural references. The rules of this language-game are inconsistent and constantly mutate, forcing a constant re-evaluation of the text and it's author, "a collapse of unification through multiplicity"(iii) as Nezavanova puts it. Remaining an anonymous, multiple and antagonistic persona allows Nezavanova to avoid becoming too cosy in the art world, making alliances and friends there who might limit the definition of what she is doing by grounding it in the specific cultural context of art.

Auto-illustrator's support mailing list performs a similar function. From reading the traffic on the list it seems that many people using it are ignorant of the fact that it is art programmed by an artist. The pseudo-corporate identity of the software company "Sinewave" maintains this ignorance with the use of more generative code. When someone emails the support list, a Perl script written by Adrian Ward generates a random identity, name and job title that he then uses to answer the query.

For example, a recent query to the list asked if there was going to be a Mac OS X port of the software. A fictional character called "Jon Tippecanoe" in the "Os X port development department" gave a short, rude answer and signed off "I suppose I'm going to get fired now".

In both these "software support" projects the tone switches constantly from helpful to playful to insulting, mirroring the unpredictable, conflicting processes of trying to make something with the software.

By constantly switching between collaborative and antagonistic attitudes towards the user, these pieces of software shift fluidly between being useful and useless, gratifying and frustrating, funny and scary.

This ambiguity also extends to media that is produced using the software. On the one had it is often aesthetically horrible, a partly random product made without clear intention, but on the other hand it is the result of a fascinating and unique collaborative process between the artist-programmer and the software user.

Perhaps the most challenging uncertainty for users of Auto-illustrator and b1257+12 is that while the software is not definitely useful, it does definitely cost money. The piracy protection on both pieces of software is far more sophisticated than many large commercial software packages, requiring server-based registration keys that are verified each time the software is installed on a new computer. A full version of b1257+12 costs $96.69 and Auto-illustrator costs about $50 to register (unregistered copies eventually expire and according to Adrian Ward, will start behaving very badly, imposing algorithmic authorship more assertively).

The behaviour of software companies, jealously guarding copyright of their expensive products is not usually associated with artistic approaches to making software, but in these cases it does work both conceptually (forcing the user to pay clearly defines this as "not just art"), and economically, allowing them to maintain financial independence from corporate, art-world or state funding.

Using these tools, and certainly programming them does address problems of authorship and intellectual property that many artists have struggled with when using software and digital media, by making the struggle for authorial control very explicit.

By maintaining a delicate ambiguity these projects avoid definition as conceptual artistic interventions or as straight forward efficiency enhancing software tools.

Notes

(i) A comprehensive list of art browsers with links would be useful so please email saul@twenteenthcentury.com with any I miss out and I'll add them. This list includes interfaces to the web as a whole, not to a single web site or database (e.g. Rhizome's alt.browser series).

I/O/D's Webstalker: www.backspace.org/iod

Jodi's Wrongbrowser: www.wrongbrowser.com

Nullpointer's Web Tracer: http://www.nullpointer.co.uk/-/webtracer/

Mark Napier's Shredder: www.potatoland.org/shredder

Tom Corby and Gavin Baily's Reconnoitre: http://www.reconnoitre.net

Andi Freeman's Earshot (feat. Jason Skeet), Funksolegrind and notScape:
http://www.deepdisc.com/ns/, http://www.deepdisc.com/earshot/

Mark Dagett's Browser Gestures:
http://www.flavoredthunder.com/dev/browser-gestures/

Many examples generated by the International Browserday competition:
www.waag.org/browser

(ii) Later projects and collaborations by the makers of the Web Stalker with Mongrel have resulted in projects like the Linker (http://www.linker.org.uk/) and it's use in training workshops (www.mongrelx.org) that fulfil this definition of "not just art" far more successfully. However an analysis of the tactics employed by Mongrel in using the Linker for "arts education" and their methods of avoiding that institutional dead-end go far beyond the scope of this text.