Jim Andrews / THE <BODY> OF NET ART

The <body> of net art can be conceptualized in many ways, but what I'd like to do here, with an eye to the future, is consider it as a kind of architecture.

But, first, let's note that while the term "net art" has widely been identified as "web art," i.e., art that you experience on the web in a browser, it should be thought of more broadly as digital art in which an internet connection is necessary though the work itself may or may not be located on the net. For instance, an installation piece in a gallery may use a voice recognition system that feeds the recognized language to Google image search in order to display or process images associated with the language spoken by gallery participants. Here, the internet connection is crucial, but the work is not one of web art which, currently, is typically viewed in a browser with the viewer sitting in front of a computer using a mouse and keyboard for input.

Most of what has been known as net art could as well be experienced on a CD without the computer having an internet connection. I don't mean to argue that such work should not be considered net art; often, the milieu and subject matter are net-oriented, and it's on the net for reasons deeper than the net being a convenient, cheap way to publish. However, I do think that the edge of net art will continue to be strongly involved with trying to do funky shit with an internet connection. Why? That's a crucial distinguishing characteristic of net art versus other types of art, and it's usually the case that the unique characteristics of a medium or a person or whatever are crucial to their identity, to the key strengths and weaknesses.

What funk is there to be accomplished with an internet connection? Works of net art are typically interactive, and the quality and possible range of the computer's response can be greatly enhanced using the internet. Many different businesses and organizations on the internet are starting to make "web services" available to programmers. These "web services" typically allow programmers to tap into things like Google text search or image, sound, video or news search; or allow programmers to tap into things like dictionaries, thesauri, auto-translators – and anything else we can use and search on the web. There are also web services such as those available at pandorabots.com that allow programmers to construct a chatbot. A chatbot is an application that you converse with. Antoine Schmitt's piece *Puppet President* (puppetpresident.net), for instance, uses web services from pandorabots.com.

Many of these services are relatively sophisticated in their ability to analyze and respond to language – and this is sure to be a type of web service that grows more flexible and wide-ranging in the services it offers. For instance, when you do a Google image search, the images are usually relevant to the search language. And when you use Schmitt's puppetpresident.net, the responses are occasionally surprisingly good. There are, of course, vistas of room for improvement, but interesting art does not have to depend on the quality of response of the web services but, instead, has to be able to do interesting, enlightening, telling, possibly useful things with what is available, possibly even undermining the "intelligence" of the web service response. It isn't the mere fact of using a particular web service that makes a net work interesting (or not), but what is done with it.

In any case, we can conceive of these web services as part of the <body> of a work of net art. A work of net art has access to any number of web services that allow it to retrieve media and also analyze and respond to the language or other actions of the viewer. Web services are not just a type of memory for the brain of a work of net art, but provide some of the intellectual functions. Typically, analysis of language is a difficult programming proposition and thousands of people have spent their entire life's work on the matter, making small but significant contributions to this field. Increasingly, net art will take advantage of that work via web services.

Web services give programmers access to millions of texts, images, sounds video, and other media such as Flash files. And usually web services provide access to these media in such a way that a database is queried for appropriate results, as is the case concerning Google image search.[1] When net art takes advantage of web services, the art becomes less about traditional media-making as creating an entity that makes telling use of the media and language analysis at its fingertips.

Using web services is not a shortcut to creating art, however. The interest of the piece as a work of art is going to be in *what is done* with the web service, *not the simple fact of its use.* The life of a work of art has more to do with its liveliness, its provocations, and its ability to connect than with specific technical achievements.

Works of software art *always* make use of services available not remotely over the net, but rather from the operating system of the local computer. These range from tapping the computer's graphic display device to its input/output devices (such as the mouse and keyboard) to using the browser and/or installed plugins such as Shockwave or Flash or Quicktime, speech recognition software, or virus scanning. These respond more quickly than web services, typically, since the query and response do not have to travel over the net, and are common services required by the general user rather than being more specialized services available over the net. Also, local services generally do not draw on the world's media stores but are relatively confined in what they can offer in terms of media and artful analysis. In any case, the local services are definitely a part of the net art <body> architecture. They offer swift response, a factor which is sometimes crucial in computer art, and customized use of all of the resources of a contemporary computer.

We see in things like Google Desktop,[2] an application that extends Google's search function to local files, how web services and local services can sometimes be merged in such a way as to make indistinguishable one's own computer capabilities or storage capacity from those of remote computers on the net. Networks connect computers in ways that vastly increase the capabilities of each of the computers on the network and, concomitantly, the capabilities of the people using them. Since McLuhan, we have thought of technology as an extension of the body, the senses, and our cognitive processes, and that is true, in spades, of computer networks.

Another extension of the <body> of net art is its facilitation of growth and change over time according to what different people do with it collectively. For instance, a Wiki[3] allows people to edit the pages of a website. Ideally, this results in stronger pages, over time, despite common fears of inaccuracy in public knowledge. In any case, the edited page is stored permanently not on the reader's machine but on the server from which the page was retrieved. This is not so much a "mental function of the brain of the work of art" so much as a type of memory available to the work of art. And this sort of memory would be important to any possibility of the work of art coming to learn anything over time. If it can't change over time, it also can't learn. Neither can it become a kind of collective work of art.

Client-server architecture can be used not only to save information from different contributors (and present viewers with all the different contributions) but also to facilitate and coordinate live communication between participants as in chat applications, where what you type is sent first to the server and then to the other people in the chat. Sometimes all the server does is let people establish a direct connection, bypassing the server, with other people involved in the communication (such as in many P2P applications).

I suppose you can think of the server as a type of web service. But, typically, web services do not store any significant information from their clients. They receive requests, process the request, return the results, and that's the end of the transaction. Whereas client-server relationships usually involve some storage on the server of information that is crucial to either later connections or to current connections by other people.[4]

Another type of memory that works of net art can access is local memory. For instance, when you are working with Word and you save your document, you can go back to it later and continue from where you left off. The document usually isn't saved remotely but just on your computer. You may share it later with others, but local memory allows you your own individual experience and work without having to share it with others. Similarly, when you play a game on the computer, you usually can continue from where you left off because the game state is saved to your local machine. Concerning works of art, this sort of local memory allows the experience of the work to be cumulative over time, rather than singular and finite. Each time the artwork is engaged, the experience is personalized to the user's particular interaction.[5]

Finally, there is the executable work of art itself, the program that uses the web services, client-server storage and/or processing, the local memory – and it is hoped, does interesting things with these parts of the <body> of net art. It is a piece of software. It is possible it may not, in itself, contain much media at all but may rely on the web services and memory for its media. It is an entity itself that experiences and processes these things, or allows us our own fresh experience and thoughts on these matters. It is a type of animism or artificial life. Although the life of a work of art is the life of art, finally; it is lively to a human audience or it is nothing at all.

What I've tried to do here is look at the main types of resources and storage that contemporary net art has available to it. The notion of a networked application is not a new one but there are not many that can be considered dynamite works of net art. It takes longer for the artistic imagination to acclimatize to new media than it does to create the technology behind it. Net art often requires significant skills in programming. Few contemporary artists have the necessary training. It will be interesting to see how the next generation of poet-programmers and artists influence digital culture.

It seems likely that computing devices will continue to proliferate within all machinery and become far more portable than they are now. Desktops have been replaced by laptops but the future probably includes devices constituted of a pair of glasses, two super-sensitive glove-like sensors, and possibly a microphone. We will be connected to the net on any stroll through the neighborhood, and objects will transmit their data and methods via the glasses we wear. The <body> of net art will eventually involve our own bodies in their connection to the rest of the network architecture.

NOTES

[1] Some examples of use of Google in net art: *Epiphanies* by Christophe Bruno (iterature.com/epiphanies); *Fields* by Christophe Bruno (iterature.com/fields); see Bruno's site iterature.com for many others; *dbcinema* is in-progress (Andrews) at vispo.com/dbcinema; Douwe Osinga's work at (douweosinga.com/projects/googlehacks) uses a variety of web services.

[2] desktop.google.com

[3] Perhaps the most significant Wiki is wikipedia.org; for more Wikis, google the term.

[4] Some works that involve client-server architecture: *Panel Junction* by Andy Deck (artcontext.org/act/05/panel); *Participatory Poem,* Andrews, (within a larger work called *On Lionel Kearns* at vispo.com/kearns); *Granular Synthesizer* by Chris Savage (japanesefreeware.com/granular); Gary Rosenzweig's *Gamespark* (gamespark.com/game.php?lobby).

[5] *Arteroids* (Andrews, vispo.com/arteroids) allows you, while in "play mode," to create and save your own texts. To do this, click "edit" while in "play mode." This takes you into "Word for Weirdos."