artport

http://artport.whitney.org

Christiane Paul

Comisaria Adjunta de Arte
de Nuevos Medios,
Museo Whitney de Arte Americano
Adjunct Curator of New Media Arts,
Whitney Museum of American Art

artport, un sitio web diseñado como portal principal de arte de Internet y como galería online para obras de arte digital creadas por encargo, fue lanzado por el Museo Whitney de arte Americano de Nueva York el 1 de marzo de 2001. artport proporciona, tanto recursos extensos de net art, como acceso a obras de arte originales creadas específicamente para el sitio y encargados por el Whitney.

artport consta de cinco áreas:
Una sección de entrada que archiva las páginas creadas por artistas a los que mensualmente se invita a realizar pequeñas obras de entrada para artport. Esta sección contiene enlaces a la web del artista y a sus principales proyectos. De modo que el archivo de la página de entrada funciona como una base de datos de proyectos de net art creados desde los comienzos del arte destinado a la red.
Una sección de exposición, donde se accede a muestras actuales de net art / arte digital y al archivo de exposiciones pasadas, como los proyectos de la Bienal Witney de Arte de Internet.
Un archivo de recursos con enlaces a organizaciones de nuevos medios, a galerías virtuales, a exposiciones de net art de todo el mundo, a festivales y a publicaciones sobre net art. Este archivo está en continuo desarrollo. Constantemente se agregan nuevas organizaciones y recursos.
Un área de colección, que archiva los trabajos de net art y arte digital que posee el Museo Whitney, como The World's First Collaborative Sentence, de Douglas Davis. Fue la primera obra de arte de Internet adquirida por el Museo, en 1995.
Una sección de encargos, que proporciona acceso a las obras encargadas especialmente por el Whitney para artport.

artport, a website designed as a main portal to Internet art worldwide, and as an online gallery space for new and specially commissioned net and digital art was launched by the Whitney Museum of American Art in New York on March 1, 2001. The site provides both a comprehensive resource of net art and access to original art works created specifically for the site and commissioned by the Whitney.

artport consists of five areas:
A gatepage section that archives the splash pages created by artists who are invited on a monthly basis to make a small artwork as a gateway the artport site. The gatepages contain links to the respective artist's site and most important projects, so that the gatepage archive functions as a database of net art projects created since the beginning of Web-based art.
An exhibition section, where current net art / digital arts exhibitions are accessible and past exhibitions, such as the Whitney Biennial Internet art projects, are archived.
A resources archive with links to new media organizations and virtual galleries on the Web, net art exhibitions worldwide, festivals, as well as net art publications on the Web. This archive is continually evolving as new organizations and resources are being added.
A collection area that archives the works of net art and digital art in the Whitney Museum's holdings, such as Douglas Davis' The World's First Collaborative Sentence, the first work of Internet art acquired by the Whitney Museum in 1995.
A commissions section, which provides access to artworks commissioned by the Whitney specifically for the artport site.

Idea Line

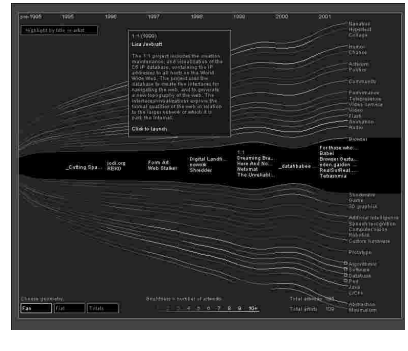
El primer proyecto encargado para artport (Idea Line, de Martin Wattenberg) fue lanzado en el otoño de 2001. Idea Line (una base de datos y una línea temporal visual de obras de arte de la red) está diseñado para demostrar tanto la variedad de temas como de tecnologías y de medios que el arte de la red ha estado utilizando, como la relación de cada obra con un panorama más amplio de aproximaciones diversas. La línea temporal (la visualización de una base de datos de proyectos de net art creados desde 1995 hasta la actualidad) tiene forma de abanico de hilos luminosos. Cada hilo corresponde a un tipo particular de arte o tecnología. El brillo de cada hilo varía en función del número de obras que contiene cada año. De modo que uno puede observar el flujo de las diferentes de líneas de pensamiento en un momento concreto. Las líneas se despliegan para revelar títulos de obras y dar acceso a la información disponible sobre ellas. Además enlazan con las obras mismas.

La base de datos de Idea Line contiene más de 200 obras realizadas por algo más de 100 artistas. Se envió una invitación a varios foros de arte de red como petición pública para contribuir en esta base. También se agregaron datos sobre muchas obras populares o influyentes que no habían sido incluídas en las respuestas. Es posible enviar e-mails a una dirección especialmente habilitada para recibir información sobre obras de net art que deseen incluirse en el proyecto.

Idea Line

The first project commissioned for artport, Idea Line by Martin Wattenberg, was launched in the fall of 2001. The Idea Line -- a database and visual timeline of net artworks -- is designed to show both the variety of themes, technologies, and media that net art has been using and the relation of each artwork to the larger tapestry of all these diverse approaches. The timeline -- a visualization of a database of net art projects that have been created from 1995 until today -- is arranged in a fan of luminous threads. Each thread corresponds to a particular kind of artwork or type of technology. The brightness of each thread varies with the number of artworks that it contains in each year, so that one can watch the ebb and flow of different lines of thought over time. The lines open up to reveal titles of artworks and access information about them as well as the artworks themselves.

The database behind the Idea Line contains more than 200 artworks by over 100 artists. An invitation to contribute to this database was sent out as a public request to several net art forums. In addition, data on many popular or influential artworks that were not covered in the responses were added. Information about net artworks can be submitted to the project by sending an e-mail to a designated address.



CODeDOC

Un segundo encargo, publicado el 16 de septiembre de 2002, fue la exposición CODeDOC. Esta muestra revisa proyectos de “software art” centrándose en “la parte de atrás” del código y comparándola con “la fachada” de las obras (el resultado del código) ya sea éste una representación visual o un proceso más abstracto de comunicación. Una docena de artistas cifraron una instrucción específica en el lenguaje de su elección. Se les pidió que intercambiasen el código entre sí para comentarlo. La instrucción era “conecta y mueve tres puntos en el espacio”. Obviamente podía interpretarse de manera literal o abstracta. Escogí esta orden por su simplicidad pero les permití introducir cierto grado de complejidad. Del mismo modo que la introducción de un tercer protagonista en una historia permite ir más allá de las posibilidades de semejanza/alianza o dicotomía que se pueden establecer entre dos personajes. El “núcleo” del código (comúnmente llamado “principal”) no debía sobrepasar los 8KB, como un documento de texto bastante corto. Los resultados de la programación no son visibles hasta que no se ha visto el código. Lo que los visitantes de artport encuentran en primer lugar es un documento con el texto del código. Desde él pueden lanzar “la fachada” del proyecto. La estrategia de presentación de CODeDOC se desvía deliberadamente de las formas en las que generalmente los espectadores experimentan las piezas de software art. Normalmente se presentan a las audiencias como código ejecutado,mostrando los resultados de las instrucciones escritas. En CODeDOC, la experiencia de visionado está más cerca del proceso creativo del artista. Lo que primero encuentra el público es una página con el código escrito. El título de la exposición se refiere a la reversibilidad del código que elabora un resultado visual y al documento escrito (doc) que contiene el código.

CODeDOC

A second commission, launched on September 16, 2002, was the exhibition CODeDOC, which takes a reverse look at 'software art' projects by focusing on and comparing the 'back end' of the code that drives the artwork's 'front end'--the result of the code, be it visuals or a more abstract communication process. A dozen artists coded a specific assignment in a language of their choice and were asked to exchange the code with each other for comments. The assignment was to 'connect and move three points in space,' which obviously could be interpreted in a literal or abstract way. I chose this assignment because it seemed simple, yet allowed to introduce a certain amount of complexity--in the way the introduction of a third protagonist into a story allows to go beyond the possibilities of either similarity / alliance or dichotomy that might unfold between two characters. The 'core' of the code (commonly referred to as the 'main') was not to exceed 8KB, which equals a fairly short text document. The results of the programming are made visible only after the code--what visitors to the artport site encounter first is a text document of code from which they can launch the front end of the project. The presentation strategy of CODeDOC deliberately deviates from the ways in which viewers usually experience a piece of software art, which commonly presents itself to the audience as executed code--the results of written instructions. In CODeDOC, the viewing experience is closer to the artist's creation process: what the audience encounters first is a page with the written code. The title of the exhibition itself alludes to the reversibility of the code producing a visual result and the written document (.doc) containing the code.

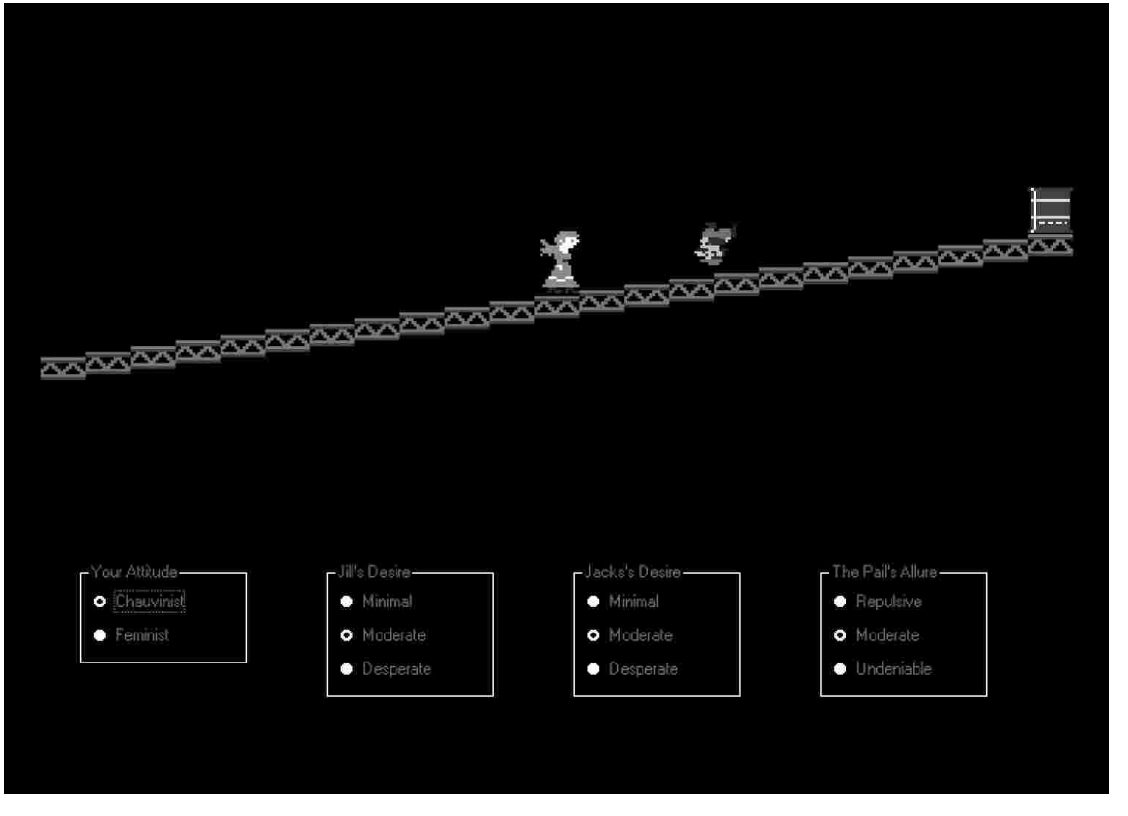
Los lenguajes en los que se escribe el código son Java, C, Visual Basic, Lingo y Perl. Obviamente, esto solamente es una selección de lenguajes de scripting y programación. El HTML (Lenguaje de Marcas de Hipertexto), la lengua de scripting en la que se basa la World Wide Web, y el flash fueron excluídos sobre todo por razones prácticas. La aceptación de estos lenguajes probablemente hubiese doblado el número de artistas participantes, convirtiendo el proyecto en algo poco manejable. Dado que el encargo imponía restricciones sustanciales en cuanto a formato y tamaño del archivo, las obras enviadas no pueden considerarse necesariamente trabajos completamente desarrollados; más bien son pequeños estudios y bosquejos que captan la visión de un artista.

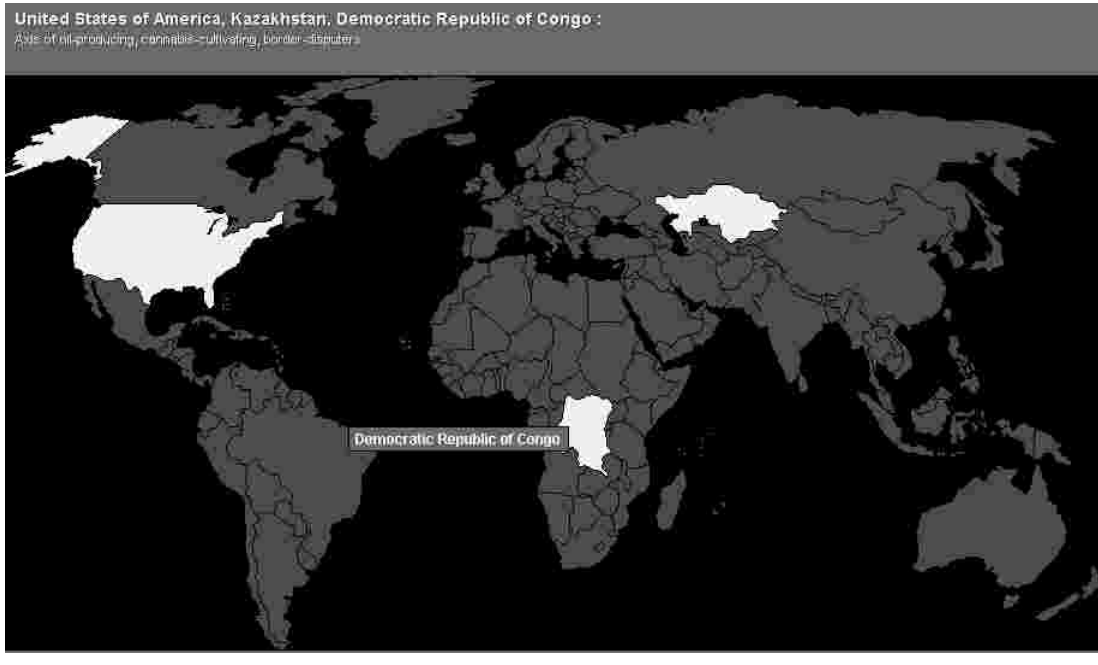
Desde el principio, CODeDOC se concivió como un experimento centrado en los procesos y no como una exposición destinada a hacer una declaración específica u a ofrecer un cierto punto de vista. Idealmente, quise plantear diversas cuestiones sobre el arte de software como práctica artística. Ni el resultado ni la recepción de este proyecto eran algo predecible. Un objetivo del proyecto fue desmitificar la noción del código como fuerza impulsora oculta y “misteriosa”, y revelar el código al espectador. Entre las preguntas que nos pareció importante tratar o clarificar estaban las siguientes: ¿Describe una cierta forma de estética el término software art en sí mismo? ¿La “firma”, “voz”, y estética de un artista se manifiestan igualmente en el código escrito que en sus resultados ejecutados? ¿Leer el código fuente mejorará la percepción de la obra? ¿Añade alguna cosa o simplemente provoca un énfasis en los “tecnicismos” innecesarios y alienantes que oscurecen el trabajo? ¿Cómo podríamos definir exactamente la relación entre “la parte trasera” del código y sus resultados?

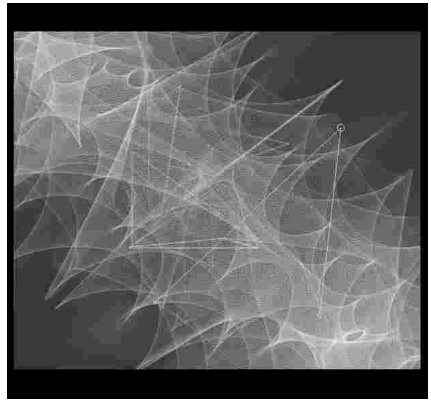
The languages in which the code is written are Java, C, Visual Basic, Lingo and Perl. Obviously, this is only a selection of scripting and programming languages. HTML (Hypertext Markup Language), the scripting language on which the World Wide Web is based, and Flash Script were excluded mostly for pragmatic reasons. The inclusion of these languages probably would have doubled the number of artists that should have been included, making the project unwieldy. Since the assignment imposed substantial restrictions in format and file size, the contributed projects can't necessarily be seen as fully developed works; rather, they are comparable to small studies and sketches that capture an artist's approach.

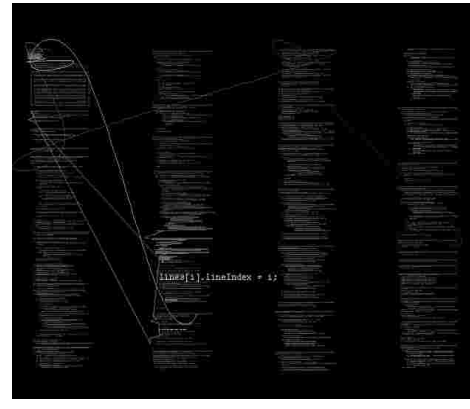
From its inception, CODeDOC was intended as a process-oriented experiment rather than an exhibition meant to make a specific statement or offer a certain point of view. Ideally, I wanted to raise questions about software art as artistic practice, and neither the outcome nor the reception of this project were easily predictable to me. One intent of the project certainly was to demystify the notion of code as a 'mysterious,' hidden driving force and to reveal the code to the viewer. Among the questions that seemed important to address or clarify were the following: does the term software art itself describe a certain form of aesthetics? do 'signature,' 'voice,' and aesthetics of an artist equally manifest themselves in the written code and its executed results? will reading the source code enhance the perception of the work? does it in fact add anything at all or just create an emphasis on 'technicalities' that is unnecessary, alienating, and obscures the work? how exactly could one define the relationship between the back end of code and its results?

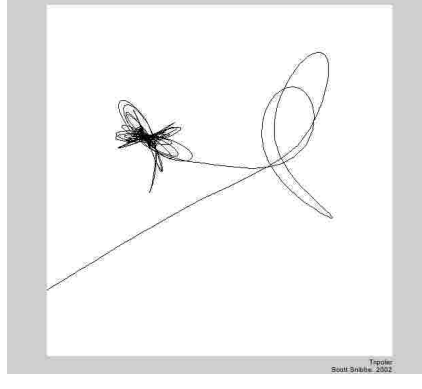
```
Sub Main()
  The_Story.Show
  While True
    If YourAttitude = CHAUVINIST Then
        If Fetch(pail, jack, jill) Then GoUpHill jack, jill
        If FellDown(jack) And BrokeCrown(jack) Then TumblingAfter jill, jack
    ElseIf YourAttitude = FEMINIST Then
        If Fetch(pail, jill, jack) Then GoUpHill jill, jack
        If FellDown(jill) And BrokeCrown(jill) Then TumblingAfter jack, jill
    End If
    The_Story.Draw
  Wend
End Sub
```

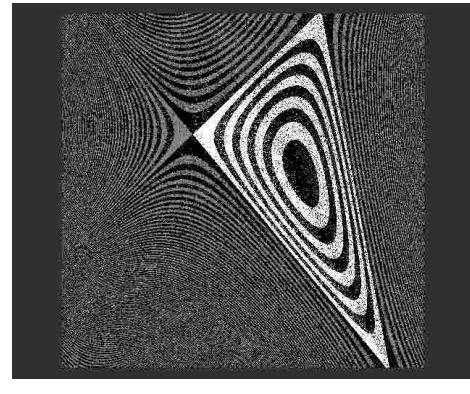



United States of America, Kazakhstan, Democratic Republic of Congo :
Axis of oil-producing, cannabis-cultivating, border-disputers
Democratic Republic of Congo




lines[i].lineIndex = i;


Tripoler
Scott Snibbe 2002



artport

Christiane Paul

La categoría de software art, usada comúnmente para software escrito por artistas, es producto de una terminología bastante borrosa. De hecho, el arte de software es mas un filtro a través del que se puede explorar un aspecto específico de la práctica del arte digital que una categoría distinta en sí misma. El software se define generalmente como un conjunto de instrucciones formales que pueden ser ejecutadas por un ordenador. Sin embargo, no hay arte digital que no tenga una capa de código y de algoritmos, un procedimiento de instrucciones formales que logren un "resultado" en un número finito de pasos. Incluso si las manifestaciones físicas y visuales del arte digital nos ocultan la capa de datos y código, cualquier "imagen digital" ha sido producida en última instancia por las instrucciones y el software utilizado para crearla o manipularla. Es precisamente esta capa de "código" e instrucciones la que constituye un nivel conceptual que conecta con trabajos artísticos anteriores, tales como los experimentos del Dadaísmo con variaciones formales y las obras conceptuales de Duchamp, Cage y Sol LeWitt basadas en la ejecución de instrucciones.

Lo qué distingue el software art de otras prácticas artísticas, es que, a diferencia de cualquier otra forma de arte visual, requiere que el artista escriba una descripción puramente verbal de su trabajo. En las formas tradicionales de arte, la "firma" y la "voz" de un artista se manifiestan en la estética de las representaciones visuales y la ejecución. Cada medio puede tener su lenguaje específico, pero en el arte digital, este lenguaje tiene una manifestación bastante más literal que figurada. En el arte del software, los resultados visuales de las obras derivan del lenguaje del código. Los lenguajes se definen por su gramática y sus reglas complejas pero, a la vez, dejan espacio para formas individuales de expresión creativa. Nuestra identidad y los papeles que desempeñamos se expresan en nuestro uso de la lengua.

The category of software art, commonly used for artist-written software, is a manifestation of fairly blurry terminology. In fact, software art is more of a filter through which one can explore a specific aspect of digital art practice rather than a distinct category in and of itself. Software is generally defined as formal instructions that can be executed by a computer. However, there is no digital art that doesn't have a layer of code and algorithms, a procedure of formal instructions that accomplish a 'result' in a finite number of steps. Even if the physical and visual manifestations of digital art distract from the layer of data and code, any 'digital image' has ultimately been produced by instructions and the software that was used to create or manipulate it. It is precisely this layer of 'code' and instructions that constitutes a conceptual level which connects to previous artistic work such as Dada's experiments with formal variations and the conceptual pieces by Duchamp, Cage, and Sol LeWitt that are based on the execution of instructions.

What distinguishes software art from other artistic practices, is that, unlike any form of visual art, it requires the artist to write a purely verbal description of their work. In traditional art forms, the 'signature' and 'voice' of an artist manifests itself in aesthetics of visuals and execution. Every medium may have its specific language but in digital art, this language has a quite literal rather than figurative manifestation. In software art, the visual results of the artwork are derived from the language of code. Languages are defined by grammar and complex rules and at the same time leave space for individual forms of creative expression. Our identity and the roles we play are expressed in our use of language.

Se puede asumir que la estética de los artistas que escriben su propio código fuente se manifiesta tanto en el código en sí mismo como en sus resultados visuales. El artista John F. Simon, Jr. ha hablado durante mucho tiempo del código como forma de escritura creativa. El código también ha sido considerado como un medio, como “la pintura y el lienzo” del artista digital. Pero el código trasciende esta comparación: permite a los artistas escribir sus propias herramientas. Para seguir con la metáfora, el medio, en este caso, también permite al artista crear la brocha y la paleta.

Como medio y práctica artística, el software art parece distinguirse de otras formas de arte como la pintura, la escultura o el cine/vídeo. En comparación con otras formas de arte visual, los artistas de software escriben instrucciones verbales para que su trabajo pueda ser ejecutado. Pueden producir desde representaciones visuales hasta procesos más abstractos de comunicación. (aunque la ejecución del código todavía requiere varios pasos de interpretación y compilación, el código en sí mismo puede ser sobre todo una notación de lógica). Hay una relación peculiar entre la “parte de atrás” del código, en su mayor parte oculta, que constituye una convergencia entre lenguaje y matemáticas y el “resultado” multisensorial que puede producir una “identidad” en el sentido de una mismidad en diversas instancias (código/resultados), que toman una forma muy diversa. Mientras que toda forma de arte se puede procesar y mediar de una manera u otra, no constituye generalmente una fusión de “materialidades” fundamentalmente diferentes (en el sentido más ámplio) como lo hace el software art. Una pintura o una escultura revelan en un alto grado las manifestaciones del proceso de su creación en el objeto acabado, por ejemplo los movimientos del pincel o los materiales, aunque el objeto artístico constituya algo mucho más grande que la suma de sus partes.

One might assume that the aesthetics of artists who write their own source code manifest themselves both in the code itself and its visual results. Artist John F. Simon, Jr. has talked for a long time about code as a form of creative writing. Code has also been referred to as the medium, the 'paint and canvas,' of the digital artist but it transcends this metaphor in that it even allows artists to write their own tools--to stay with the metaphor, the medium in this case also enables the artist to create the paintbrush and palette.

As an artistic medium and practice, software art seems to distinguish itself from other art forms such as painting, sculpture or film / video. As opposed to other forms of visual art, software artists write verbal instructions for their work that can be executed and produce anything from visuals to a more abstract communication process (although the execution of code still requires various steps of interpretation and compiling and the code itself may be mostly a notation of logic). There is a peculiar relationship between the mostly hidden backend of code--which constitutes a convergence of language and mathematics--and the multi-sensory 'display' it can produce: an 'identity' in the sense of a sameness in different instances (code /results), each of which takes a very different form yet, on one level, is one and the same. While every art form may be processed and mediated in one way or another, it usually does not constitute a fusion of fundamentally different 'materialities' (in the broadest sense) as software art does. A painting or sculpture to a large extent reveals the manifestations of its creation process in the finished object--for example, in individual brush strokes or materials--even if the art object amounts to something much larger than the sum of its parts.

Christiane Paul

En el software art, la “materialidad” de las instrucciones escritas permanece oculta. Además, estas instrucciones y notas que pueden ser activadas instantáneamente, son las obras en sí mismas. Mientras podamos afirmar lo mismo sobre un trabajo de arte conceptual que consista en instrucciones escritas, este trabajo tendrá que ser activado como evento mental o físico por el espectador y no podrá transformarse, trascender o generar su propia materialidad instantáneamente.

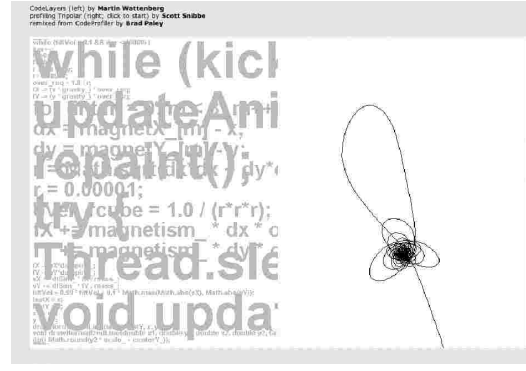
Los proyectos mostrados como parte de CODeDOC son expresiones de firmas artísticas distintas: el acercamiento conceptual al proyecto, la manera en que el código se ha escrito y los resultados producidos por él revelan mucho sobre cada autor. Algunos de los artistas de la exposición original CODeDOC interpretaron la asignación de una manera fundamentalmente gráfica, visual. Mark Napier, por ejemplo, lo hizo en en forma de "puntos elásticos" que simulan el comportamiento de resortes y masas; Martin Wattenberg y Camille Utterback en sendas exploraciones de estructuras triangulares. Hubo acercamientos más minimalistas y gráficos, como “Tripolar” de Scott Snibbe, que simula un péndulo que pivota sobre tres imanes, o “Circler” de Kevin McCoy, que explora una construcción y reconstrucción de círculos con bandas.

En "[remotion]", de Mary Flanagan, el trazado de la forma visual es utilizado para comentar la relación entre naturaleza y tecnología. Al conectar una tecnología de visión por ordenador a una webcam sin cables situada al aire libre, el software traduce el campo de la naturaleza a la tecnología de la información: materiales vivos, como hojas de árboles que crujen o ramas que se agitan, son seguidos en el tiempo (movimiento) y el espacio (posición) por tres pequeños puntos.

In software art, the 'materiality' of the written instructions mostly remains hidden. In addition, these instructions and notations can be instantaneously activated, they contain and--further layers of processing aside--are the artwork itself. While one might claim that the same holds true for a work of conceptual art that consists of written instructions, this work would still have to be activated as a mental or physical event by the viewer and cannot instantaneously transform, transcend, and generate its own materiality.

The projects featured as part of CODeDOC are expressions of distinct artistic signatures: the conceptual approach to the project, the way the code has been written, and the results produced by it reveal a lot about the respective artist. Some of the artists in the original CODeDOC exhibition interpreted the assignment in a predominantly graphic, visual way--Mark Napier in the form of "springy dots" that simulate the behavior of springs and masses; and Martin Wattenberg and Camille Utterback in their respective explorations of triangular structures. More minimalist, graphical approaches are Scott Snibbe's "Tripolar," which simulates a pendulum swinging above three magnets, or Kevin McCoy's "Circler," exploring the construction and reconstruction of circles with sprites.

In Mary Flanagan's "[remotion]," the tracing of visual form is used to comment on the relationship between nature and technology. Connecting computer vision technology to a wireless, outdoor webcam, the software translates between the terrain of nature and information technology: living materials--such as rustling tree leaves or waving branches--are traced in time (movement) and space (position) by three small points.



Otros proyectos eligieron una perspectiva más conceptual conectando puntos de la red global de internet o del mundo. Sawad Brooks creó la portada de un periódico "globalcity" partiendo de las páginas web nytimes.com, guardian.co.uk y asahi.com.

"El sentido de la vida expresado en siete líneas de código", de Maciej Wisniewski es un reloj vivo, no interactivo, o un reloj de sol que conecta cambios en la luz del día con el mundo físico y con un cómputo del tiempo y de la historia basados en la localización geográfica. Golan Levin planteó una sátira interpretando la infame creación de un "eje del mal" por George W. Bush como “un [acto] imaginativamente formal, geométrico que tuvo el efecto de erigir un triángulo monumental, virtual, de proporciones globales". Levin creó una herramienta online para conectar tres países cualquiera en el globo y encontrar semejanzas entre ellos utilizando bases de datos públicas internacionales. El proyecto permite a los usuarios crear sus propios "ejes" revelando concordancias entre los países seleccionados.

"Jack y Jill" de John Klima trata explícitamente el lenguaje del código como narrativo, recreando la nana del mismo nombre:

If Fetch(pail, jack, jill) Then GoUpHill jack, jill
If FellDown(jack) And BrokeCrown(jack) Then TumblingAfter jill, jack

El código conecta los tres “personajes” de la historia: Jack, Jill y el cubo (pail), con diversos comportamientos como el deseo de Jack y de Jill o la fascinación del cubo, que determinan su interacción. El componente visual de la pieza evoca los anales de la historia del juego (un componente muy importante en toda la obra de Klima) usa elementos de Donkey Kong como entorno para la historia y hace actuar a la Princesa y al Mario del juego original, como Jill y Jack respectivamente.

Other projects took a more conceptual approach by connecting points in the global network of the Internet or the world. Sawad Brooks created a "globalcity" newspaper front page from the web sites of nytimes.com, guardian.co.uk and asahi.com, and Maciej Wisniewski's "The Meaning of Life as Expressed in Seven Lines of Code" is a non-interactive living clock or sundial that connects changes in daylight to a computation of time and history based on geographical location and to the physical world. Golan Levin took a satirical approach by interpreting George W. Bush's infamous creation of an "Axis of Evil" as "an imaginatively formal, geometric [act], which had the effect of erecting a monumental, virtual, globe-spanning triangle": creating an online tool for connecting any three countries on the globe and finding similarities between them on the basis of international public databases, the project allows users to create their own "axis" by revealing commonalities between selected countries.

John Klima's "Jack & Jill" explicitly treats the language of code as a narrative by retelling the nursery rhyme of the same name:

If Fetch(pail, jack, jill) Then GoUpHill jack, jill
If FellDown(jack) And BrokeCrown(jack) Then TumblingAfter jill, jack

The code connects the story's three 'characters'--Jack, Jill and the pail--through various behaviors, such as Jack and Jill's desire and the pail's allure, which determine their interaction. The visual component of the piece refers to the annals of gaming history (a strong component in Klima's body of work) by using elements of the game Donkey Kong as the environment for the story and 'casting' Jill as the Princess and Jack as Mario from the original game.

Christiane Paul

Una perspectiva totalmente diferente del lenguaje, en este caso su abuso, se revela en“What you see is what you get” ("lo que ves es lo que consigues") de Alex Galloway, que coloca instrucciones ilegales (como un gusano de e-mail o una carta usada para el fraude por correo) en las categorías de “punto de entrada”, “punto de desaparición” (por ejemplo un puerto de scanner) y “punto sin retorno” (por ejemplo una bomba de bifurcación). El proyecto, obviamente, no permite su ejecución directa.

Una exploración auto-reflexiva del código es “CodeProfiles” de W. Bradford Paley, que rompe los límites entre el la “parte de atrás” y la “fachada”. "CodeProfiles" es un software que muestra su código subyacente y se comenta a sí mismo moviendo tres puntos en el "espacio del código". Una línea blanca sigue el código en el orden en que fue escrito por el artista; una línea ámbar sigue el código palabra por palabra como alguien podría leerla; y una línea verde muestra cómo el ordenador lee y ejecuta el código.
Hay un elemento procesal intrínseco al software art que permite la reconfiguración y la expansión. Como forma de comentar los proyectos, los artistas comenzaron a “remezclar” su trabajo, aplicando su propio código a otros proyectos o combinando secciones de código en una nueva obra. Trabajando con obras de CODeDOC, Brad Paley usó su "CodeProfiler" con los trabajos de Martin Wattenberg y Scott Snibbe para perfilar sus códigos. Wattenberg contraatacó creando su propio software “perfilador”, “Code Layers” que permite a los espectadores ver el programa leyendo el "CodeProfiler" de Brad mientras se perfila. Mientras que "CodeProfiler" muestra el movimiento y el ritmo secuencial de la ejecución del código, el “Code Layers” de Martin explora las operaciones paralelas que suceden en un momento dado.

CODeDOC es un intento de observar más de cerca el proceso de esta práctica artística en particular y de cuestionar los parámetros de la creación artística. En el mejor de los casos, la exposición puede hacernos entender de otra manera lo que rodea la construcción y la percepción del software art. Espero que las piezas creadas para este proyecto continúen dialogando.

A completely different take on language, in thiscase its abuse, unfolds in Alex Galloway's "What You See Is What You Get," which arranges illegal instructions, such as an e-mail worm or letter used for e-mail fraud, into the categories of 'Entry Point,' 'Vanishing Point' (e.g. a port scanner) and 'Point of 'No Return' (e.g. a fork bomb). The project obviously does not allow for its direct execution.

A self-reflexive exploration of code is W. Bradford Paley's "CodeProfiles," which collapses the boundaries between front end and back end. "CodeProfiles" is a software that displays its underlying code and comments on itself by moving three points in "code space": a white line traces the code in the order it was written by the artist; an amber line traces the code word by word as someone might read it; and a green line shows how the computer reads and executes the code.
Intrinsic to software art is a procedural element that allows for reconfiguration and extension, and, as way of commenting on the projects, artists started to 'remix' their work, applying their own code to other projects or combining sections of code into a new project. As a comment on other works in CODeDOC, Brad Paley turned his "CodeProfiler" on the works of Martin Wattenberg and Scott Snibbe in order to profile their code. Wattenberg countered by creating his own 'profiling' software, "Code Layers," allowing viewers to watch his program reading Brad's "CodeProfiler" as it profiles itself. While "CodeProfiler" shows the sequential movement and rhythm of the code's execution, Martin's "Code Layers" explore the parallel operations happening at any given moment.

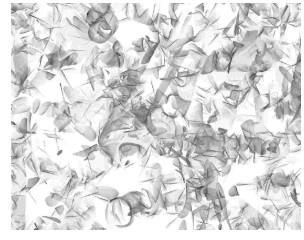
CODeDOC is an endeavor to take a closer look at the process of this particular artistic practice, and to ask questions about the parameters of artistic creation. At best, the exhibition can raise some awareness surrounding both the construction and perception of software art, and I hope that the pieces created for this two-part project will continue to contribute to an ongoing dialogue.

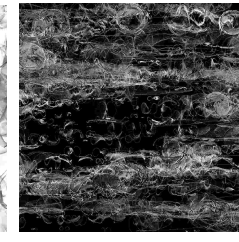
{Software} Structures
El tercer encargo, {Software} Structures de Casey Reas (con Roberto Hodgin, William Ngan, Jared Tarbell), publicado en agosto de 2004, continuó la investigación de software art. Inspirándose en los murales de Sol LeWitt, {Software} Structures explora la importancia del arte conceptual para la idea del software art. Reas tradujo tres dibujos de Lewitt a software y luego los modificó. Después de trabajar con planos de LeWitt, creó tres estructuras únicas (descripciones del texto que contornean relaciones dinámicas entre elementos) que fueron puestas en ejecución: 26 piezas de software derivadas de las estructuras textuales se cifraron para aislar diversos componentes, incluyendo la interpretación, los materiales, y el proceso. Reas invitó a tres personas a interpretar y a poner en marcha una de sus estructuras de software ("una superficie llena con cien círculos. Cada círculo tenía un tamaño y dirección diferentes, pero se movía con la misma lentitud"). Esta estructura fue ejecutada en tres entornos de programación separados (Java, flash, C++). Finalmente, el proceso de realización de la estructura se reveló documentando su evolución desde las primeras líneas del código hasta el trabajo terminado. Para cada una de las puestas en práctica, se puede ver el software, el código fuente, y los comentarios de los artistas.

artport continuará ofreciendo proyectos encargados para el sitio web. Pronto alojará Breakup Visualization System de Golan Levin (con Kamal Nigam), una visualización interactiva de información que utiliza los datos sacados automáticamente de weblogs y sigue las vidas amorosas de un millón de adolescentes americanos. La visualización se actualiza con información fresca diariamente y permite a los visitantes navegar, buscar, clasificar, filtrar y comparar visualizaciones regidas por datos de miles de relaciones románticas. ■

{Software} Structures
The third commission, {Software} Structures by Casey Reas (with Robert Hodgin, William Ngan, Jared Tarbell), which launched in August 2004, continued the investigation of software art. Inspired by Sol LeWitt's wall drawings, {Software} Structures explores the relevance of conceptual art to the idea of software as art. Reas implemented three of Lewitt's drawings in software and then made modifications. After working with the LeWitt plans, he created three unique structures -- text descriptions outlining dynamic relations between elements -- which were then implemented: 26 pieces of software derived from the textual structures were coded to isolate different components, including interpretation, material, and process. Reas invited three people to interpret and implement one of his software structures ("A surface filled with one hundred medium to small sized circles. Each circle has a different size and direction, but moves at the same slow rate."). In addition, this structure was implemented in three separate programming environments (Java, Flash, C++). Finally, the process of realizing one structure is revealed by documenting its evolution from the first lines of code to the completed work. For each of the implementations, you may view the software, source code, and comments by the artists.

artport will continue to feature commissioned projects at the site. Coming soon will be Breakup Visualization System by Golan Levin (with Kamal Nigam), an interactive information visualization that uses data automatically scraped from web logs and plots the romantic lives of a million American teenagers. The visualization is updated with fresh information daily, and allows visitors to browse, search, sort, filter, and compare data-driven visualizations of thousands of romantic relationships. ■





"Not Just Art" -- from Media Art to Artware

Christiane Paul

Conservadora Adjunta de Artes
de Nuevos Medios,
Museo Whitney de Arte
Adjunct Curator of New Media Arts,
Whitney Museum of American Art

In its standard usage, the term "media" is used for communication, information, or entertainment systems in their various forms and, until the later part of the 20th century, was mostly associated with broadcasting media -- one-to-many distribution systems. Since the 1950s, the spectrum of "older" media -- most notably print, radio, film, and television -- was broadened with the advent of technologies such as color television, news satellites, video recorders and tapes, videophones, cable television, laser techniques, as well as computer and networking technologies. It is the latter two forms of media, in particular, that finally enabled a substantial shift to a many-to-many communication and distribution system that diminished the boundaries between sender and receiver or producer and consumer.

While the potential of this shift to many-to-many distribution networks was recognized much earlier and was indeed one of the dreams of video art, the "new" (digital) media of the late 20th and 21st century finally allowed a fairly fluent and broader implementation of this many-to-many model. Digital media have facilitated a form of "re-mediation" in the sense of a reconfiguration of both the possibilities of media systems and the connections between art and media, in particular. While these reconfigurations have been an ongoing process throughout the history of media arts, the digital medium and its networked characteristics have proposed new models for questioning and restructuring these systems. A significant subcategory of new media arts essentially consists of alternative models for media systems and tools -- "artware" that is "not just art" but a proposal for the restructuring or critique of existing media systems.

Artists have always used and reflected on the technology of their time and the so-called "media arts" have a history that is as long as the history of media itself. Artists also started to expand the possibilities of the one-to-many broadcasting media at a time when the concept of many-to-many distribution systems was hardly recognized by the public in general. In the 1960s, Max Neuhaus defined new arenas for music performance by staging sound works in public arenas and experimenting with networked sound as a form of "virtual architecture." In the first installment of his project Public Supply (1966), he established a connection between the WBAI radio station in New York and the telephone network, implementing a 20-mile aural space around New York City, where participants could intervene in the performance by making a phone call. When Sony portapaks became available in the 1970s, artists and activists used this portable recording power for establishing alternative media networks, addressing issues of documentation and representation in the context of control over media distribution. However, the attempt to establish distribution systems for the public at a larger scale ultimately failed. Apart from the fact that media systems can only be reconfigured with the combined creative endeavors of many individuals, earlier technologies such as video also still required far more complicated processing and distribution facilities than today's new media do. Collaborative experimentation with independent media systems and with creativity and automated also tools can be found in the art-machines and robotics of the 1960s. This experimentation is also embedded in the larger context of value creation within the art market -- an issue that was explicitly taken on by art movements of the 1960s and 70s, which tried to move art outside of an institutionalized context.

Using "new technology" such as video and satellites, artists in the 1970s began to experiment with live, networked performances that anticipated the interactions now taking place on the Internet and through the use of streaming media. The focus of these projects ranged from the application of satellite technology for extending the mass dissemination of a television broadcast to the aesthetic potential of video teleconferencing and the exploration of real-time virtual space that collapsed geographic boundaries. David Ross, among others, has examined the parallels between the video art practices of the 70s and networked media art in the 90s, which were both striving to establish new cultural systems of exchange.[1]

Many of the discussions surrounding the reconfiguration of media systems and questions about agency within these systems take place within the new media arts practice that is concerned with the creation of artware. The inherent hope and promise here is that software production can be seen in the broader context of cultural production or, as Pit Schultz has put it, "that writing code has more meaning than making a program run or crash or sell".[2] Software always has to be seen as cultural construct, and the creation of artware addresses this construct from various angles, including the enhancement or re-engineering of existing software products; the creation of alternative, community-driven platforms of exchange; and the examination of agency, autonomy, or political agendas in software.

The creation and analysis of today's artware and Do-It-Yourself networks is embedded in the larger context of what is referred to as "software art." The term software art should be understood here as a specific filter applied to artistic practice that involves the writing of software, rather than a clearly distinguishable category of new media arts.

The jury of the Read_me 1.2 festival[3] broadly defined software art as art based on code as formal instructions, or art offering a cultural reflection of software -- definitions that cover a broad territory. If one takes a look at the subcategories listed on the runme software repository's site[4] one encounters a landscape that may be fairly confusing in its topography but nevertheless makes important distinctions and can still be roughly summed up under the above mentioned definitions. Labels such as algorithmic appreciation, generative art, code poetry, data transformation, as well as digital folk and artisanship (e.g. ascii art and screen savers) arguably seem to put an emphasis on the aesthetics of formal instructions. On the other hand, classifications such as existing software manipulations (cracks and patches or plug-ins) or political and activist software (e.g. cease-and-desist-ware and software resistance) as well as software tools point to the role of software art as critical reflection of software's cultural status, its encoded political or commercial agenda. It is mostly in this context where the development of artware unfolds. The "art aspect" manifests itself in the writing of software -- or the re-writing / re-engineering of existing software -- as an act that examines the underbelly, inscribed aesthetics, and agenda of the original construct and thus opens it up to discussion. The inherent hope and promise here is that software production can be seen in the broader context of cultural production or, as Pit Schultz has put it, "that writing code has more meaning than making a program run or crash or sell."[5]

"Not Just Art" -- from Media Art to Artware

Christiane Paul

Software always has to be seen as cultural construct, and the creation of artware addresses this construct from various angles, including the enhancement or re-engineering of existing software products; the creation of alternative, community-driven platforms of exchange; and the examination of agency, autonomy, or political agendas in software.

Obviously, artists today are working with different technologies than they did 30 or 40 years ago, but one still has to ask the question, is there anything that fundamentally distinguishes today's endeavors to reconfigure media and establish new cultural systems of exchange from previous ones? I would argue that a fundamental difference lies in the nature and specifics of the technology itself. Previous media, such as radio, video or television mostly relied on a technological super-structure of production, transmission, and reception that was relatively defined. The modularity and variability of the digital medium however constitutes a far broader and more scattered landscape of production and distribution. Not only is there a plethora of softwares, each responsible for different tasks (such as image manipulation, 3D modeling, Web browsing etc.) but due to the modularity of the medium, these softwares can also potentially be manipulated or expanded. As a result, there are several potential points of intervention for artistic practice. As Saul Albert's diagram [Fig.1] shows, intervention could take place between software and media producers or producers and consumers. In the following, I will discuss some examples of artware or artistic tools that intervene at different points.

Browser and Search Engine Reconfigurations

Over the past decade, numerous art projects have either intervened with existing browsers and search engines or created applications that expanded these tools' functionalities.

The British group I/O/D single-handedly established browser art as artistic practice with their WebStalker [6] [Fig. 2], an application that allows users to draw "frames" in a blank window and select information they would like to display in them -- for example, a graphical map of the site that presents all its individual pages and the links between them; the text from a URL and the source code of the HTML page; a "stash" of URLs users would like to save. Although the WebStalker didn't display graphics, it expanded the functionality of existing browsers in a way that questions the paradigms of the conventional information display and Internet 'architecture.' In his essay "A Means of Mutation," Matthew Fuller, "A Means of Mutation" [7] I/O/D's Matthew Fuller described the Web Stalker as "not just art" -- a form of cultural practice that could have an impact outside of the relatively narrow confines of the art world or even sustain itself (although the latter did not quite happen).

While the WebStalker engages notions of the browser as culturally coded construct, one also shouldn't neglect its distinct aesthetics and their art-historical references. In his essay "Visceral Facades: taking Matta-Clark's crowbar to software,"[8] I/O/D's Matthew Fuller establishes a connection between the WebStalker's approach to information architecture and American artist

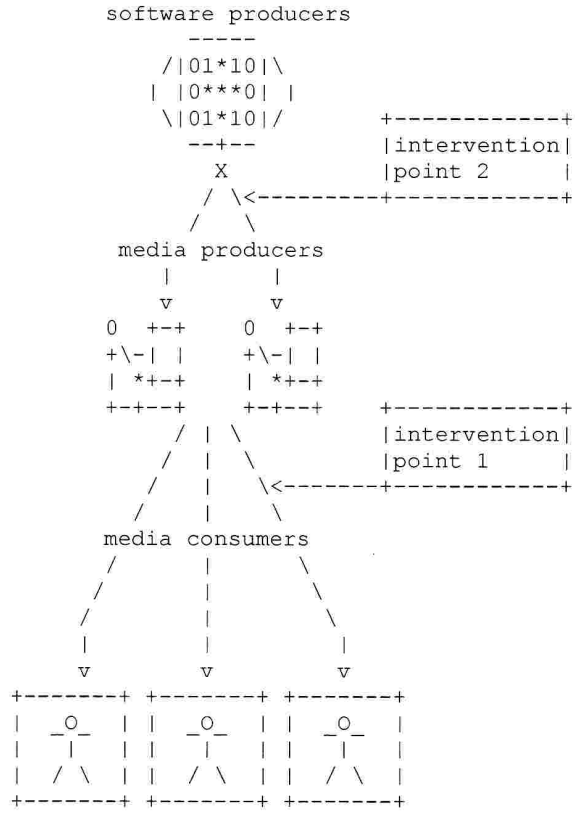


Fig. 1. Diagram from Saul Albert, "Useless Utilities" in Auto-Illustrator Users Guide (Signwave: London, UK, 2002)

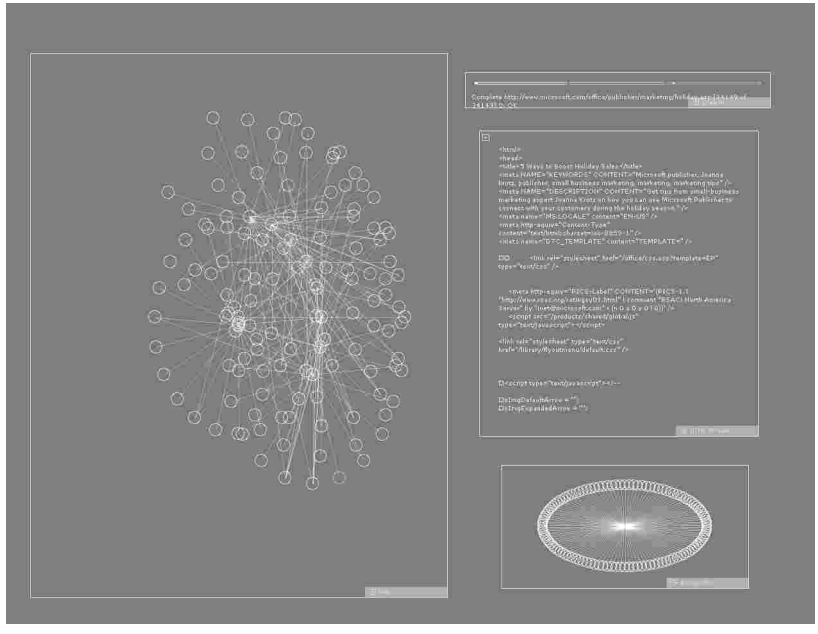


Fig. 2. I/O/D, WebStalker. Screenshot

Gordon Matta-Clark's technique of literally "splitting" the existing architecture of buildings, an application of formal procedures that would result in a revelation of structural properties. Matta-Clark's as well as the WebStalker's "deconstructionism" and "anarchitecture" are as much statements against certain social conditions as they are aesthetic acts oscillating between reconstructions of the destroyed and destructions of closure.

While different in its approach, Maciej *Wisniewski's netomat*™ [9] [Fig. 3] -- a meta-browser abandoning the page format of traditional browsers and treating the Internet as one large database of files that can be searched by typing in keywords or questions -- would fall in the same category of alternative browsers. Using an audio-visual language designed specifically to explore the unexplored Internet, netomat™ reveals how the ever-expanding network interprets and reinterprets cultural concepts and themes and takes visitors for a ride into the Internet's "subconscious." Netomat ultimately came closer to the concept of "not just art" since it now exists as a company (Netomat Inc.) that turned the original Web browser's underlying software and technology into a product and alternative model for communicating online.

Browser art has become a broad field of artistic exploration that has produced many well-known interventions, among them Jodi's Wrongbrowser, Nullpointer's *Web Tracer or Mark Napier's Shredder and Riot* [10], all of which address specifics of the browser in very different ways (from aesthetic to political). An example of the expansion of browser functionalities would be Martin Wattenberg's and Marek Walczak's *Wonderwalker* [11], a project commissioned by the Walker Art Center and alluding to the Wunderkammer or cabinet of curiosities. [Fig. 4] The Wonderwalker allows users to create a shared, public map of favorite sites and thus turns the Web browser's bookmark function into a participatory space for exchanging sites of interest.

The area of search engine reconfigurations has been equally prolific and has produced a wide array of projects.[12] Andy Deck's *Culture Map* [13][Fig. 5], for example, gives its users a comparative view of the contents of the Web according to certain topics. The project allows users to choose from up to 32 categories (such as News, Shopping, Economy, Arts) and then uses data from different search engines to determine the "scale" of the topics according to the predominance of the term in Web pages. The resulting visualization assigns a colored region to each of topic, its size varying according to the occurrence of the term. Culture Map -- a piece of "meta-cartographic information art," as Deck calls it -- critically examines how people find information through the categorical entry points and key themes of search engines and throws light on the bias of the engines themselves.

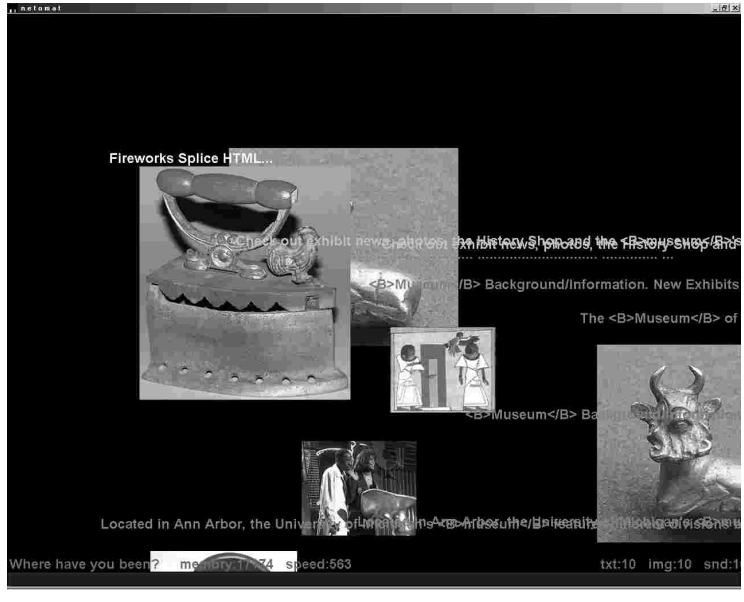


Fig. 3.
Maciej Wisniewski's netomat™. Screenshot

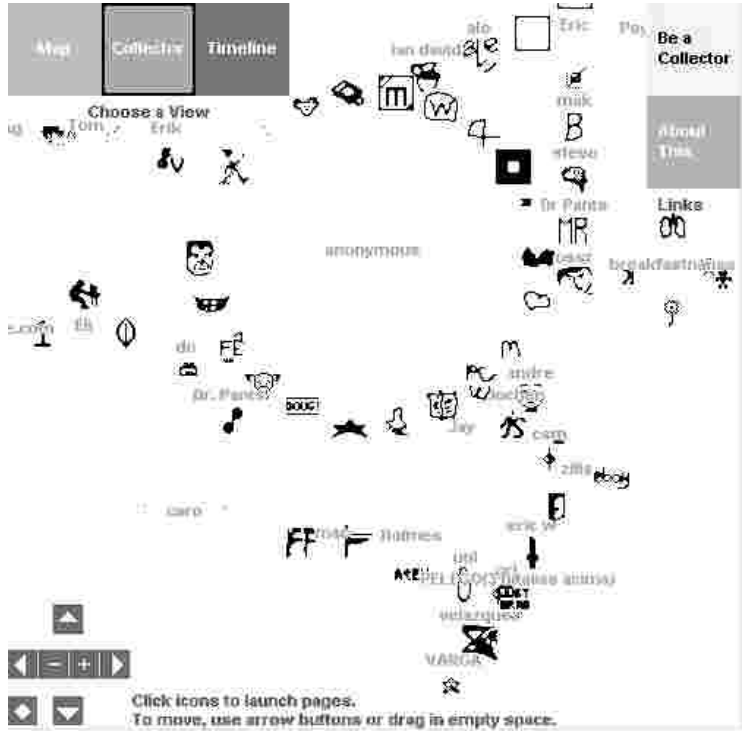


Fig. 4.
Martin Wattenberg's and Marek Walczak's Wonderwalker. Screenshot

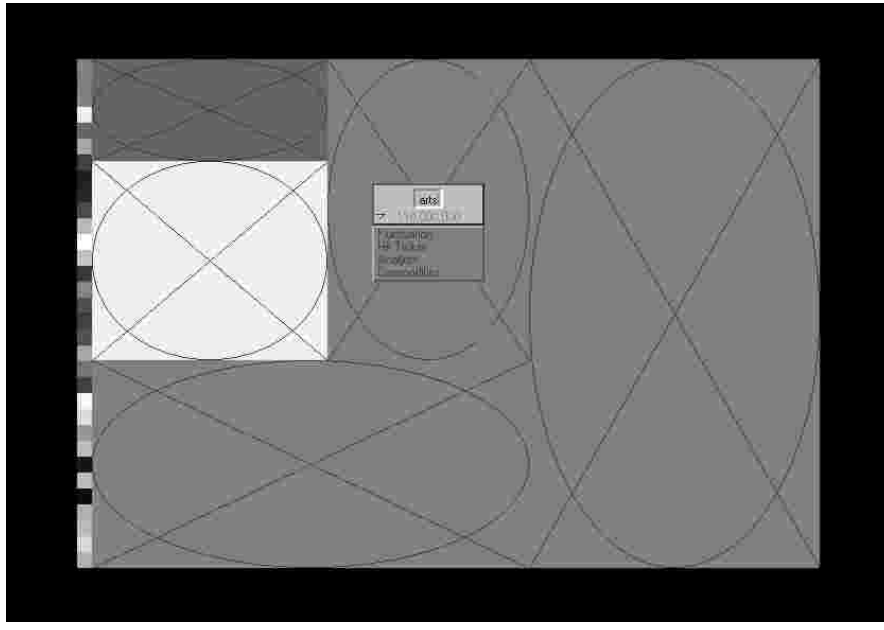


Fig. 5.
Andy Deck's Culture
Map. Screenshot

A search engine intervention that focuses on privacy rather than data analysis is offered by TraceNoizer[14] , a project by LAN, a varying group of students, media workers, artists and designers. Billing itself as "Disinformation on Demand," the project addresses issues surrounding the "databody" -- the accumulated traces of information that people leave on the Web through their homepages, institutional affiliation, participation in conferences etc. While some people consider their databody a useful necessity that provides them with exposure (for example regarding their research and publications), others regard it as an unwelcome nuisance that makes them vulnerable to intrusion by marketers etc. TraceNoizer is a tool that uses disinformation as a strategy for data protection. The project finds data related to a person's name and / or e-mail address (for example, information connected to people with a similar name and address) and clones it into a number of homepages in a fully automated process. These homepages are then automatically uploaded to servers that provide free web space and thus become accessible through search engines and other Web surfers. The result is a multitude of data clones for one person that make it impossible to arrive at valid information about the individual in question. As a combined search, analysis, and publication tool, TraceNoizer misleads through a process of open cloning.

Media Production Tools

Apart from artistic expansions of the software utilities that are used by Internet "media consumers" on a daily basis (Web browsers, search engines), there also is a large body of artistic tools that establishes a framework for the production of media content. Some of these works explicitly allude to or transform the standards of commercial software for drawing or image manipulation. Andy Deck's *Open Studio*[15] , for example, a multi-user online "drawing board," offers its user a palette of options for "spray-painting" with their mouse. What most radically distinguishes this application from any of its commercial counterparts is the ability to draw together with multiple people in remote locations in real-time. In most cases, the focus of users shifts from the creation of their respective painting to "responding" to the other people simultaneously occupying the space. The experience becomes closer to a live, graphic jam and constitutes a break with the usual context of computer drawing applications.

One of the most well known artistic graphic design applications remains Adrian Ward's Signwave Auto-Illustrator 1.1[16] , developed on the basis of his earlier project Autoshop 1.0 [17].
Both applications obviously allude to and parody Adobe™ Photoshop™ and Illustrator™, respectively. Auto-Illustrator is an explicit statement about the conventions and standardization of commercial graphic design applications and at the same time explores the beauty and elegance of generative graphic design. As opposed to most other artistic tools, Auto-Illustrator deliberately follows a commercial model, being sold as a software package with a limited edition user license -- a fact that both highlights its validity as a tool and suggests an alternative model for selling art.

Using an industry standard-interface, Auto-Illustrator uses a familiar tool palette -- including a pencil, brush, oval, rectangle, and text tools -- but extends the regular options by offering sliders that can automatically create a rectangle in a shabby or precise childish or adult design, for example. The filters allow users to generate 1970s boxes or architectures; parody sportswear logos; or insert instant Murakami eyes (with the "SuckMyPixel" filter). Using the "Bug" tool, one can place bugs (with modifiable behaviors) into the document that will crawl around, drawing lines behind them. [Fig. 6] In an ironic way, Auto-Illustrator "illustrates" the limits of commercial design applications by frustrating the expectations one might have of them and thus highlighting the standardized operations on which they are based.

While automating creativity in a generative process, Auto-Illustrator explores the interrelationship and agency of the author / user / software at any given point. As Ward explains, the Artificial Intelligence routines (previously employed in Autoshop) "randomly" seed incoming data but then filter this data based on rules of logic established by the author. While the computer uses random data to determine certain factors, these come into play only when they produce suitable results. Since the role randomness plays in software projects is often overrated and misrepresented, it is important to note that -- as Ward puts it -- "a computer can only move data about. It cannot -- under any circumstances -- generate a truly random number by itself." [18] For Ward, a system's ability to feed data back into itself -- thereby becoming chaotic, complex, and dynamical -- is comparable to the unpredictability of creativeness itself.[19] In works such as Auto-Illustrator, the concept of "feedback" becomes a complex interplay between the author, user, and software. The agency of the code may be considerable but it is certainly not the only party "speaking."

Ward states that "While some consider technology totalitarian, others forge ahead by expressing their creativity as technological tools, treating technology not as a system of control, but a system of growth." [20] Without neglecting the amount of control that technology in general can exert, programming certainly offers unprecedented possibilities for shaping technology. Code is not only an artistic "medium" comparable to paint or clay, it also allows artists to write their own paint brushes and chisels.

A considerable body of "media production tools" has been developed within the field of music, DJing and Vjing and often takes the form of interactive sound processors (for example, Netochka Nezvanova's b1257+12[21] or musical "instruments," such as John Klima's software glasbead. John Klima, glasbead. [22] Informed by multi-user environments, gaming, and file-sharing, glasbead is a multi-user collaborative musical interface and instrument that allows players to import and share sound files and create a myriad of soundscapes. The interface consists of a rotating, circular structure with stems that resemble hammers and bells. Sound files can be imported into the bells and are triggered by flinging the hammers into the bells. [Fig. 7] While glasbead creates a contained world where sounds and visuals enhance each other, it allows up to 20 players to remotely "jam" with each other. The project was inspired by Hermann Hesse's novel Das Glasperlenspiel (The Glassbead Game, published in English under the title Magister Ludi), which applies the geometries of music to the construction of synesthetic microworlds.

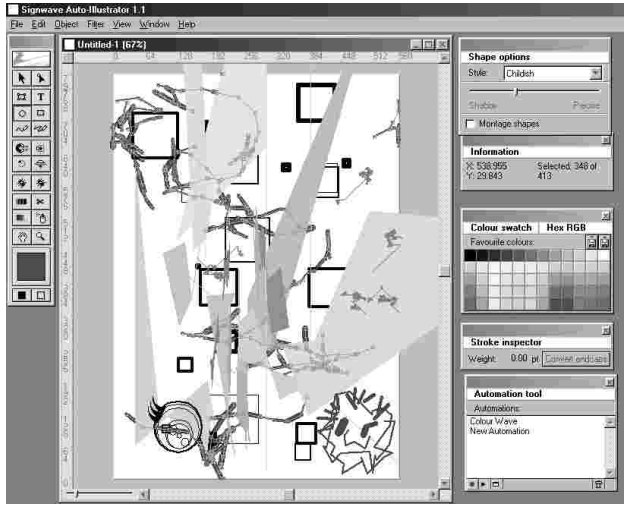


Fig. 6. Adrian Ward, Signwave Auto-Illustrator 1.1. Screenshot: Generated architecture (green) and 1970s boxes; instant Murakami eye (bottom left); shabby childish oval (bottom right); red bugs.

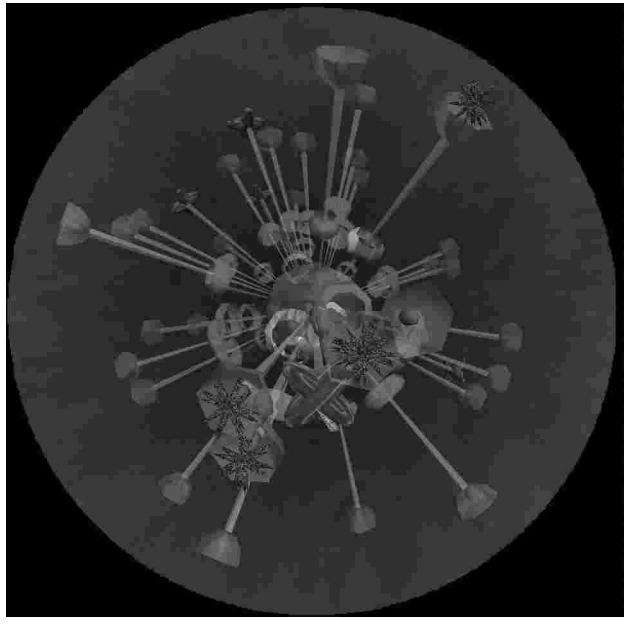


Fig. 7. J ohn Klima, glasbead. Screenshot

A wide area of artware consists of "social software" -- tools that are aimed at providing platforms for community-based exchanges and publishing. An example of this type of project would be Nine(9) by the British collaborative Mongrel. Nine(9) is a continuation of Mongrel's project Linker[23] and was created by Mongrel member Harwood while he was artist-in-residence at the Waag Society Amsterdam. The project is an open-source software structure that allows individuals and communities to "map" their experiences and "social geographies." Nine(9) consists of a server-based application that can incorporate 9 groups x 9 archives x 9 maps = 729 collective knowledge maps. An important part of the project as "social software" is an ongoing dialogue between users and programmers in order to transcend standardized social relations. In a very different way and context, both Nine(9) and Auto-Illustrator play with limitations -- in structure or functionality, respectively -- to test and explore possibilities of software.

Other projects, such as Liken by criticalartware (Ben Syverson, Jon Cates, Jon Satrom and Blithe Riley -- core developers) investigate community-driven interfaces for social software.[24] Liken is a Web interface (with various different manifestations) to criticalartware's database of shared resources that present themselves as self-connecting nodes to which users can contribute. The pathways connecting the nodes change on the basis of usage, with more "traveled" paths growing stronger and paths attracting less interest fading away. Criticalartware's approach is that of hybridization, a self-reflexive crossbreeding of interfaces and connected threads that becomes a social document in itself.

An excellent portal for exploring free software tools for collaborative networking and media production is the DIVE CD-ROM, which was created by <KOP> (Kingdom of Piracy) and commissioned by the VirtualCentre-Media.Net and FACT, UK. The CD-ROM includes projects such as Mongrel's Nine(9), Radioqualia's Frequency Clock, and LAST.FM, a peer-to-peer network for streaming customized selections of music. [25]

The "re-mediation" unfolding in the above-mentioned projects takes the form of models for mediated exchange that transcend simplistic receiver / transmitter structures. These models explore inherent possibilities of media systems and offer alternatives outside of the media industry. The new "art media" may not radically redefine connections between art and media but they certainly have opened the field of artistic engagement and agency. Whether alternative media systems and artware projects will have a mass appeal and profound impact on existing structures remains debatable. While they are mostly community-driven, they certainly can make use of a distribution system of unprecedented scale, and there is no doubt that art projects have been noticed by the industry. The rise of Linux (a topic in itself) is an indication that open-source systems can offer alternatives that are taken seriously and implemented on a larger scale. Even if the impact of artistic media reconfigurations remains limited, they are a much needed "reality check" -- a critical examination of today's media and proposal for alternatives. ■

[1] -David Ross, "Net.art in the age of digital reproduction" ("Art and the age of the digital"), lecture at San Jose State University (Cadre), March 2, 1999: http://switch.sjsu.edu/web/v5n1/ross/index.html. Edited version reprinted in *Camerawork: A Journal of Photographic* Arts, Vol. 26 No 1, Spring / Summer 1999.
[2] - Pit Schultz, "QuickView on Software Art," http://runme.org/project/+quickview
[3] - http://www.m-cult.org/read_me/
[4] - http://www.runme.org; developed by Amy Alexander, Florian Cramer, Matthew Fuller, Olga Goriunova, Thomax Kaulmann, Alex McLean, Pit Schultz, Alexei Shulgin, and The Yes Men
[5] - "QuickView on Software Art," http://runme.org/project/+quickview
[6] - I/O/D, *WebStalker* (http://www.backspace.org/iod/)
[7] - Matthew Fuller, "A Means of Mutation" (March 1998), http://www.backspace.org/iod/mutation.html
[8] - Matthew Fuller, "Visceral Facades: taking Matta-Clark's crowbar to software," http://www.backspace.org/iod/Visceral.html
[9] - Maciej Wisniewski, *netomat™*, http://www.netomat.net
[10] -Jodi, *Wrongbrowser*, http://www.wrongbrowser.org
Nullpointer, Web Tracer, http://www.nullpointer.co.uk/~/webtracer
Mark Napier Shredder http://www.potatoland.org/shredder/ / *Riot* http://www.potatoland.org/riot/
[11] -Martin Wattenberg & Marek Walczak, Wonderwalker http://wonderwalker.walkerart.org/#
[12] - See for example Julia Page and Tommy Walker, "The Art of the Engine," http://people.mills.edu/jpage/
[13] - Andy Deck, *Culture Map*, http://artcontext.org/cultmap/index.php
[14] - LAN, *Tracenoizer*, http://www.tracenoizer.org/
[15] - Andy Deck, *Open Studio*, http://draw.artcontext.net/cgi/index.cgi?lang=english
[16] - Adrian Ward, *Signwave Auto-Illustrator*, http://www.auto-illustrator.com; an upgrade to 1.2 was amde available in 2003.
[17] - http://www.signwave.co.uk/products/autoshop
[18] - Adrian Ward, "How I Drew One of my Pictures" in *Auto-Illustrator User*s Guide (Signwave: London, UK, 2002), p.73, 72;
[19] - Ibid., p. 73
[20] - Adrian Ward, excerpt from 4x4: *Life and Oblivion: Generative Design in Auto-Illustrator Users Guide* (Signwave: London, UK, 2002), p. 96
[21] - http://en.wikipedia.org/wiki/Netochka_Nezvanova http://framework.v2.nl/archive/archive/node/actor/default.xslt/nodenr-66625
[22] - John Klima, glasbead (1999/2000), http://www.glasbead.com
[23] - Mongrel, *Nine*, http://9.waag.org; *Linker*, http://www.linker.org.uk
[24] - Criticalartware, *Liken*, http://www.criticalartware.net/lib/liken/
[25] - <KOP>, DIVE, http://kop.fact.co.uk/DIVE/cd/dive/index.html; Radioqualia, Frequency Clock, http://www.frequencyclock.net; Michael Breidenbruecker, Felix Miller, Martin Stiksel, Thomas Willomitzer, LAST.FM, http://last.fm

Christiane Paul
Conservadora adjunta de artes de nuevos medios en el Museo Whitney de Arte Americano y directora de Intelligent Agent (http://www.intelligentagent.com), una organización de servicios y recursos de información dedicada al arte digital. Ha escrito extensamente sobre arte de nuevos medios, net art, arquitectura de la información, hipermedios e hiperficción, y sus artículos se han publicado en revistas como Sculpture, Leonardo, e Intelligent Agent, su libro Digital Art (parte de la serie World of Art, Thames & Hudson, Reino Unido) fue publicado en julio de 2003. Actualmente está corrigiendo una antología, Curating New Media. Imparte clases en el departamento de artes informáticas MFA de la Escuela de Artes Visuales de Nueva York y ha dado conferencias sobre arte y tecnología internacionalmente.

En el Museo de Whitney, se encargó de la muestra "Data Dynamics" (2001), que trataba del mapeo de datos y el flujo de información de Internet, de la selección de net art para la bienal Whitney 2002, así como de la exposición en línea "CODeDOC" (2002) para Artport, el portal online del Museo Whitney para arte de Internet del que ella es responsable. Otros trabajos como conservadora incluyen "The Passage of Mirage" (Museo de Arte de Chelsea, Nueva York, 2004); "Evident Traces" (Festival Ciberarts Bilbao, 2004); "eVolution -- the art of living systems" (Art Interactive, Boston, 2004); "CODeDOC II" (Ars Electronica, 2003); la exposición del décimo aniversario del New York Digital Salon (NYC, 2003); "Mapping Transitions" en la Universidad de Boulder, Colorado (2002); "Re-Media" (Fotofest, Houston, Texas, 2002); y una selección de net art para"Evo1" (Galería L, Moscú, octubre 2001).

Christiane Paul ha participado en numerosos paneles de nuevos medios y ha presentado conferencias por todo el mundo. Sus intervenciones incluyen el simposio "Media Art - Art Media," ZKM (Centro para la Cultura y los Medios), Karlsruhe, Alemania; Forum ARCO 2004 y 2005, Madrid, España; el Museo Tate, Lodres; el museo de arte contemporáneo (MACBA), Barcelona, España; el Festival Cyberarts de Boston; la Real Academia de Bellas Artes de Estocolmo, Suecia; la Real Academia de Bellas Artes de Copenhague, Dinamarca; la conferencia anual de la Asociación de Arte de la Universidad (NuevaYork); la Cumbre Internacional sobre Multimedia e Internet (Abu Dhabi, EAU); the International Summit on Multimedia and the Internet (Abu Dhabi, UAE); invenção thinking the next millennium (Sao Paulo, Brasil); consciousness reframed 2 (CAiiA, Gales, Newport, RU); y la Governor's Conference on the Arts (San Francisco).

Christiane Paul
Is the Adjunct Curator of New Media Arts at the Whitney Museum of American Art and the director of Intelligent Agent (http://www.intelligentagent.com), a service organization and information resource dedicated to digital art. She has written extensively on new media, net art, information architecture, hypermedia, and hyperfiction, and her articles have been published in magazines such as Sculpture, Leonardo, and Intelligent Agent,. Her book "Digital Art" (part of the World of Art Series by Thames & Hudson, UK) was published in July 2003. She is currently editing an anthology on Curating New Media. She teaches in the MFA computer arts department at the School of Visual Arts in New York and has lectured internationally on art and technology.

At the Whitney Museum, she curated the show "Data Dynamics" (2001), which dealt with the mapping of data and information flow on the Internet and in the museum space; the net art selection for the 2002 Whitney Biennial; as well as the online exhibition "CODeDOC" (2002) for artport, the Whitney Museum's online portal to Internet art for which she is responsible. Other curatorial work includes "Evident Traces" (Ciberarts Festival Bilbao, 2004); "eVolution -- the art of living systems" (Art Interactive, Boston, 2004); "CODeDOC II" (Ars Electronica, 2003); the New York Digital Salon's 10th anniversary exhibition (NYC, 2003); "Mapping Transitions" at the University of Boulder, Colorado (2002); "Re-Media" (Fotofest, Houston, Texas, 2002); and a net art selection for "Evo1" (Gallery L, Moscow, October 2001).

Christiane Paul has participated in numerous panels on new media and presented at conferences worldwide. Her speaking engagements included the symposium "Media Art - Art Media," ZKM (Center for Culture and Media), Karlsruhe, Germany; ARCO Forum 2004 and 2005, Madrid, Spain; the Tate Museum, London; the Museum of Contemporary Arts (MACBA), Barcelona, Spain; the Boston Cyberarts Festival; the Royal Academy of Arts, Stockholm, Sweden; the Royal Academy of Arts, Copenhagen, Denmark; the annual College Art Association conference (New York); the International Summit on Multimedia and the Internet (Abu Dhabi, UAE); invenção thinking the next millennium (São Paulo, Brazil); consciousness reframed 2 (CAiiA, Wales, Newport, UK); and the Governor's Conference on the Arts (San Francisco).