

Running head: Schema for Born-Digital Media Artworks

An Evaluation Schema for Born-Digital Media Artworks?

Examining the Preservation of Internet-based Art

Tabea Lurk

Bern University of the Arts / AktiveArchive

Fellerstrasse 11

CH – 3027 Berne (Switzerland)

tabea.lurk@bfh.ch

Tel.: +41 (0)76 372 2131

Jürgen Enge

Karlsruhe University of Arts and Design

Lorenzstrasse 15

D-76135 Karlsruhe

juergen@hfg-karlsruhe.de

Tel.: + 49 (0)721 8203 2312

Abstract

This paper is dedicated to the conservation of network-based artworks. It conveys aspects of a currently evolving art-technological understanding of the nature of net art. Excerpts of an evaluation code of practice for analyzing and documenting net art are introduced. They list the core elements and the communication structures which must be sustained in order to achieve long-term preservation.

The applied methodology combines the conservatory documentation practice and experience in recording artworks with technical descriptions of the artworks' functionalities and communication processes. Art-historical understanding is taken into consideration to localize the internal work logic. It links the artistic concept with the technical settings.

An Evaluation Schema / Manual / Code of Practice for Digital-Media Artworks?

Examining the Preservation of Internet-based Art

Introduction

The susceptibility of net art to technical failures and the dependency on specific system environments encourage the consideration of new approaches for lasting (long-term) preservation strategies. The question is what *sustainability* means in a field, in which both the physical conditions for displaying the artwork and the environmental context of the Internet are continually changing? How may distributed artworks which lack a clear physical structure be documented, archived, and preserved?

We started out with technological considerations connected to a specific Internet-based artwork as singular unit, but found during the course of our research the focus shifted towards a more general understanding of cultural values and parameters, which are embedded within the technological environment of the artwork. As far as the mode of preserving digital objects correlates with specific object classes (Hunter & Choudhury, 2003, p. 5), we narrowed the scope of our research down to the digital components and their technical real-time environment. We are convinced that authenticity – as a matter of accuracy (Laurenson, 2006, p. 6) - and meaning rely very often on specific technological settings. Nevertheless it is important to keep in mind that “[d]ifferent authenticity criteria” as Rothenberg & Bikson (1999, p. 7) claim, “produce different preservation requirements that can be met by alternative technological approaches.” Against this background we want to answer the question: To what extent is it possible to sustain the digital components of the computerized artworks?

We have structured this essay in two sections. First the phenomenon of net-art will be introduced as both the subject of art-technological research and an object to be preserved. After

commenting on basic structures and characteristics of net-art the theory of encapsulation is applied. The second part focuses on the three main areas of operation: the client part, the server part and the network communication. These observations touch on impending threats and suggest art-technological procedures, which might help to estimate the operational expenses in terms of preservation. These issues will be illustrated with artistic examples. Finally summing up this article an excerpt of our evaluation chart for sustaining originally born digital artworks is introduced.

Even though the text primarily focuses on technical concerns it is geared towards persons not educated in the information sciences. Practical advice is given to assist with detecting common failures, estimating effort and developing guidelines for preservation strategies. We want to call attention to common drop-outs and degradation patterns of net art. Some approaches are offered for a better understanding of the meaning of network-based artworks.

Objectives

For illustration of a widely-used practice in net art restoration, we start with the following example: Reinhard Storz, the curator of the Swiss net art node of *X-Cult.org*, adapted one of his collective net art projects called *The RAM Show* (1999) to a current technological standard (Figure 1) in 2004. He ‘translated’ the former HTML-programming to a Flash application. The complex memory-piece contains file-based multimedia content, audio and video recordings and animated images. An important element is the interactive navigation - offering cross-links and dynamic narratives. The maintenance of the artwork in Flash complied with the author’s interest in keeping the audio-visual appearance of the piece alive.

This method shows vividly that very often the immaterial components or rather (in)visible processes are primarily observed as a generator of surface effects. The digital setting

of the artwork seems less important than the permanent availability of the artwork and the artistic concept. The coding is understood as *variable* (Hanhardt, 2003). Modifications on this level seem less evil than a drop-out. Changes are meant to be inevitable.

The convoluted design of *The RAM Show* seems quite common for that period of net art and multitudes of examples exist. Often the original programming is interrupted or even disrupted by current browsers. Scripting and / or source code are frequently displayed incorrectly. In addition the manner and speed of Internet access have changed frequently. Due to this many artists and net pioneers decided to update their artworks continually and adjusted the coding to the modified context. Olia Lialina reflects this aspect for example in regard to the artwork *My Boyfriend came back from the War* (2008). The artist lists the history of different editions or versions of the work and provides access via a web-archive.

Scope of Research

By means of bug fixing and updating their pieces many artists unwittingly abandon historic information and connotations. Due to this fact our research tries to sensitize the stakeholders to the importance of maintaining a certain historic accuracy – or authenticity and integrity as Laurensen (2006) claims. Digital components which are proven important within the artwork should not simply be replaced by others.

The altered elements may contain important cultural information. They derive from a specific technological era and are evidence of specific technical developments. The original coding and display not only provide important information on a certain technological and historical setting but also suggest clear aesthetic enunciations. One can also argue that the original setting has influenced the artistic formulation. The artwork may be dated in an iconographical way and in terms of media-archeology (Laforet, 2009, 13ff.). In a different

context the media theorist Georg Ch. Tholens (2002) introduced the idea of cultural threshold situations, which are called *media fractures* (*Medienumbrüche*). Applying this terminology to our scope of research we conclude that technically-caused failures may occur as an indicator for intentional changes within a cultural setting.

Net art – Art without Objects

Working on net art often means dealing with objects which are by their nature locally distributed and physically dissolved. The classical setting implies the storage of the source code on a server engine with the (audio-)visual surface of the artwork ‘appearing’ on a client machine in a browser, player or interpreter.

Terminology

To facilitate understanding we will call the server host of an artwork *art-server* whereas any additional web-servers are called *data providers*. Finally there are *web and news-services* which basically provide metadata information and link to related data providers in the network environment (Internet). The client provides the user interface. If special services on the art-server are required they are specified. Artworks which are located solely on community portals like Youtube or Flickr are excluded for the moment.

Shifting Work Constellations

Whereas in former times the core (content) of a web-based artwork was usually stored on the art-server today many artworks consist of framesets. The art-server provides a certain amount of instructions, which are embedded in a specific layout and executed on the client. Additionally content from external data providers is gathered and reassembled on the client in order to complete the artwork. This means that the operational logic of the artwork is often shifted to the client.

Technically speaking, the overall design (appearance) of the artwork is embedded in an action script application and downloaded by the user. The art-server generates a meta-search which points to data from external data providers. The client operates the art-script, downloads the reference data from the art-server and processes the indicated web-content. The client itself becomes *active*. This new form of activity goes beyond former browser art.

Net art is currently part of a communication process. This does not indicate that it is strictly immaterial. It is possible to identify executing files and trace the behavior of the artwork. The object status is dissolved, however.

Identifying and Encapsulating the Artwork

The *double presence* of the artwork – its' existence on the art-server and the client – is not necessarily a disadvantage. It may indeed support the preservation process, because a combination of different preservation strategies arises. Also the modular structure allows the concentration on single aspects.

As the integrity of the artwork does not only relate to its display (surface effect) but also to the programming, to preserve it the dismantling of the work-relevant components from the surrounding hardware is of major importance. As Rothenberg (2006, p. 26) explains: “The only way to guarantee that th[e] behavior is retained is to run those original programs, without rewriting them or converting them in any way.” Dislocating the artwork from the former host means replacing elements which have environmental status with digital components which meet exactly the required functionality. In order to avoid any loss of information a careful documentation is recommended. Full access to the artwork and its related elements is necessary. Components that are not accessible are considered as *environmental*.

The core operation within our research is the distinction between *work-relevant components* and *digital environment*. While the work-relevant components are defined as part of the artwork itself (because they derive from the artist), the digital environment provides a kind of digital playground for the artwork. It enables its existence.

The crucial point is now to identify *the artwork* among the dissolved assembly of machinery, different system environments and communication processes. Here the logical unit called *digital environment* becomes important. It offers a specific kind of real-time performance, which is required by software-based art (Lurk, 2008). The digital environment seems offer more stability for future application than the original hardware because it is much more adaptive (Granger, 2001). The artwork is transferred to a digital environment instead of being reprogrammed. Preferably it should not be undergoing any other changes. Lee et al. (2002, p. 98) discuss this approach under the term of *encapsulation*.

In our case studies we recreated authentic digital environments by virtualizing the host machines. Afterwards a copy of the artistic code was ported to the virtual machine (VM). The VM offers an isolated, secure environment for examining all work-relevant components and the artwork itself. It contains an authentic operating system, the artistic software and additional modules (e.g. helpers), which are required by the artwork. Technically fragile components are encapsulated in this artificial safety zone (Figure 2). The VM can be seen as a digital carrier or a transitory object, which prepares the artwork for forthcoming migration processes, additional changes may occur only on the outer boundaries of the VM.

Modes of Documentation

Prior to the porting process all metadata information has to be gathered carefully: internal work flows, internal and external interfaces and any additional elements are to be recorded. The artistic concept has to be retrieved and related technically to the work logic.

According to art-technological standards a complete technical documentation of the hard- and software components is recommended. At the same time an all-embracing technical documentation, which simply lists all hard- and software components, does not necessarily reveal the internal work logic of the artwork, especially if all components are denominated with the same priority.

In addition to a complete documentation we suggest introducing hierarchies and operations to set priorities within the registration process. Attributes characterizing specific components according to their pertinence help to reveal the core elements of the artwork. Communication processes within the art-scripts and between art-server, client and external data providers are to be analyzed in regard to the internal work logic and in relation to the artistic concept.

Our approach is based on the theory of abstraction (Figure 3). In computer sciences one deduces specific procedures and system requirements of a higher or current level out of the behavior of a lower one (Marinescu & Kröger, 2007). A strictly technical documentation might start with documenting the hardware of the machine followed by an excerpt of the hardware abstraction level. It would continue with the kernel (kernel mode drivers and microkernel), the operating system including hardware drivers etc, and might finally end up with the application level.

Usually network-based artworks are located at the application layer.

Analyzing Net Art

The suggested closer look at the application layer (called User Mode Subsystem in Figure 3) highlights software modules and scripting elements which should be preserved. We focus on three areas of operation: components required by the client, elements required by the art-server and the network communication.

The client requirements reflect the recipients' point of view. We start by characterizing different types of clients. In a second step we suggest how to chart the operational logics of the artwork. Finally core functionalities and reliabilities are described.

The considerations focusing on the art-server are the following: The operational logic on the art-server describes the interaction between different applications and internal rendering processes. Especially important is the linkage between the artists' script and the system environment of the art-server.

In order to sustain the operational logic of an artwork it seems indispensable to comprehend the communication between the client, the art-server and - when applicable - the external data-providers. The Netart Router is introduced as a software based tool, which supports analyzing processes. The observation of the network communication enables the characterization of different types/genres of net art. The over-all design of the artwork is described.

Observing Client Requirements

As far as the display of network-based artworks is realized through clients, it seems obvious to consider the requirements for preserving clients first. But what is a client? In our context the client is comprehended to be a software (tool) which engages specific services from a server. This broad definition correlates with the lexical definition of a client in computing [Wikipedia, 2008]. It meets the art-technological scope, because in the art context not only do we

have to deal with browsers, players and interpreters but sometimes also with specific hardware interfaces. Steven Greenwood for example, uses a sewing machine in his internet-installation *Woven Present* (1996), which sews white text on black ribbon. Peter Dittmers artwork *The Amme* (1992-2005) utilizes objects like a glass of milk, light bulbs and additional hard- and software-based actors.

Unlike those artworks the following considerations do not imply extensive hardware. They operate on the application layer.

Characterizing Different Types of Clients

Concentrating on the application layer of the client enables the characterization of different types of clients. In the following we distinguish between *browser-based* and *application-based* clients. Browser-based clients employ common web-browsers like Internet Explorer, Mozilla Firefox, Safari, applets (e.g. Java Web Applets), and plug-ins (e.g. Quicktime, RealMedia, Flash). Application-based clients, however, require specific software in addition to a browser, which has to be installed on the client. We think here especially of certain media players like the Adobe-Flash-Player which exist in both modes. Due to different reasons some artists prefer their work being displayed by the application instead of a browser. In those cases the application communicates directly with the network interface.

On the one hand application-based clients seem easier to maintain than browser-based ones: In general the executing files are compiled which means that all immediately required elements (or example fonts) are embedded within the executable file. The file-format is adapted to the application and should execute the artists' application naturally. To this end it seems sufficient to keep the correct type and version of the application accessible. On the other hand the reliance on the surrounding system environment (= operating system) grows according to the

increasing complexity. Very often an artwork programmed on a Mac does not run properly on a PC and vice versa - even if the required application is supposed to be available for different types of computers. Type, version and current operating system have therefore high priority.

Reliance on the runtime environment or a specific computer platform is often not that obvious, especially if the required functionality is anchored deeply within the operating system. Quite some artworks access standard widgets like control elements, check-boxes, combo-boxes, radio-buttons etc. They are usually used during the search process (e.g. Internet Explorer on Windows machines) for calculations or during upload processes. If the applied upload function varies much from the client machine and is not designed by the artist, the *look and feel* changes from client to client – according to the operating system.

The dependency of the client on a specific system environment grows with the growth of the embedded operational logic. Furthermore especially older artworks often need ancient system environments - whereas nowadays software is usually designed to operate across platforms. This leads to the conclusion that the older an application-based artwork or displaying software is, the stronger the dependence on an original surrounding results. The preservation of application-based clients has to take into consideration those aspects during the definition of the digital environment, which contains the artwork and related software modules for display. Speaking visually the outer border of the artwork is shifted.

Live executable environments like virtual machines or emulations seem more permanent in this context than static files: “For fully digital or analog/digital artworks of these kinds, whose behavior is controlled by a computer program, the safest preservation strategy seems to be run the work’s original program to ensure that the logic of its behavior is not changed” (Rothenberg, 2006, p. 25). For preserving software clients we therefore recommend the collection of out-dated

software applications and their system environments in digital carriers. Furthermore a precise documentation of the former host might be helpful in cases where the artwork is transferred to a current system.

Standard Requirements for Clients

As an interim result we can sum up that the following aspects need to be investigated:

- In cases where the artwork is displayed in a browser, the correct type and version of the browser has to be identified. If no specific information is available the date of the artwork, in combination with a browser history allows an accurate estimate.
- In addition to the availability of the required plug-ins, JavaScript-versions, rendering engines etc. have to be revised. Very often the source code of the displayed html-page contains the required information. For identifying the correct configuration of the rendering engine ACID-tests are used. They check the correct application of line breaks, accessibility of fonts, style sheets etc.
- Specific in regard to browser-based applications are the features of variable extensions (like <open new window> or the <blink->tag). Pop-up blockers, tools used to eliminate advertisements, and other filters are very often embedded in a browser. If they are active, they create some small but still not unimportant issues, as they may interfere with the immediate idea of artworks, which rely on the opening of a new window or even a sequence of windows.
- In regard to stand-alone applications the information concerning the required software is usually provided by the artist. Some artists place particular importance on using stand-alone players as their display rather than browser-based plug-ins. In addition to this explicit information the filename extension of the art-script and

the source code of the application may lead to a solution, should the information be missing.

- Congruent to the browser-displayed artworks, the correct type and version of any application also has to be identified.
- Specific external interfaces which are used as navigational tools – like remote controls (Lee, 2002), barcode readers (Enge & Vasulka, 2003/04) or specific devices manufactured by the artist are another factor.

Identifying the Operational Logic of the Client

In addition to the standard requirements for display tools the functionality of the client within the over-all design of the artwork is important. The activity or operational logic of the client is related to the *displaying mode*. Facing video based installation artwork Saaze (2009, 194) even claims that “authenticity can be explored as being done in practice” and she concludes at a certain state: “Perhaps authenticity here is best described as continuity. Unlike authenticity, continuity does not imply a point zero, a frozen moment in time to which the work of art should return. Instead, continuity suggests maintenance but also allows for flexibility and change” (Saaze, 2009, 197). Observation of the displaying mode exposes the operational logic of the artwork. It relates the clients’ engagement to the artistic work-logic.

Concerning the operational logic of the client we divide between *statically displayed* and *dynamically rendered* display: Statically displayed content is downloaded as a – more or less – self-contained file from the art-server. It is interpretable immediately by the application software or browser. Opposed to this dynamically rendered content is (re-) assembled within the client.

In general dynamical rendering processes engage intelligent operations that go beyond common browsing functions. Reference materials, for example links, are provided by the art-

server (e.g. in a XML-string) but are executed on the client. A XML-string contains information about related, digital objects like the location (host) and format as well as technical and descriptive metadata. The client then downloads items appropriate to the referred data from an external data-provider into the browser or application, and reassembles and displays it according to the artists idea (Lee, 2006; Brogle, 2004)

Determining the displaying mode of an artwork means considering the rendering potential of the client. Though the rendering potential in combination with *intelligence* are soft criteria, we suggest a graph (Figure 4) to identify the different activities on the artworks.

Whereas the x-axis presents the development from static display of content like plain html-files to dynamically rendered content, the y-axis locates the discussed types of clients: browser-based clients, application-based ones and a combination thereof.

The graph suggests an estimate of the client's reliance on its current environment. The higher the rendering activity the more the application relies on the operating system. The complexity of the internal work logic within the client grows along the implied diagonal. It is important to emphasize that in this context complexity does not have any relation to the artistic quality.

Examples I

An example for a complex application-based client is provided by Marc Lees' installation *Breaking the News. Be a News-Jockey* (2006). It uses an action-script framework that utilizes dynamic content from external data-providers. The display is designed to be in the form of a Flash-movie. This Flash-movie contains 19 different news-modules that assimilate continually changing content (Figure 5). A so-called *image-module* for example positions pictures in a grid structure; a *bubbles-module* encloses words in semi-transparent spheres, which seem to float

upwards on the screen. Continuous text is often stretched across the whole screen. Images of selected web-cams seem placed above a huge world map. It becomes obvious that the defined work logic is highly complex. Each news-module provides parameters, graphical patterns, timing and positioning for the audio-visual content.

Reliance on Environmental Objects

It is obvious that sustaining *the artwork* becomes quite difficult. We distinct once again between work relevant components and environmental elements. It helps to classify different conditions and preservation strategies. In general it can be observed that elements loaded from the art-server into the browser or plug-in belong directly to the artwork. Content downloaded from external data providers, however, obtains environmental status. As long as there is any kind of access to the art-server the data or communication between art-server and client seems manageable. Unfortunately it is not possible to guarantee the availability of environmental data long term – to correct this recently developed documentation tools offer novel conservation approaches.

Considerations on the Server Side

In relation to the server part of the artwork, we divide between server-based operations and communicative processes. Server-based operations concentrate on internal rendering processes on the art-server while communicative operations engage the external environment. We label those *context-sensitive*.

In an art-technological scope determining the server's functionality implies the identification of services which are used by the artists' software. Normally artists utilize standard or preinstalled applications that adopt the common functionality of a web-server, a streaming-

server, a mail-client etc. Whereas different services are frequently applied within one artwork, direct intervention on the design-level of common services is only rarely implemented.

Analog to the client side the dependency on the system environment has to be captured. In general the server-infrastructure allows more flexibility due to the system architecture offering a higher level of abstraction in regard to hardware and to software environment. During the course of its existence the runtime environment of an art-server is often modified or even exchanged. Changes are mostly caused by server updates. This is why the definition of the proper system environment is very important.

Different to the client part of the artwork the relation between art-script and server environment is more complex on the server. Preinstalled software-libraries and modules are often accessed directly from the artwork. Whereas artistic scripts on the client side frequently exist as compiled files (see above) the artistic application on the server engages (pre)existent operations e.g. for calculation processes. This means that it is not sufficient to store the art-script. The art-script relies on additional modules installed on the server from the archive. As far as they enable the functionality of the artwork they can be replaced manually after porting the artwork to a new server. Those algorithms are often modified in order to produce smoother effects.

To sustain the artwork, additionally to data files and scripts, which are supposed to *contain* the artwork (Rinehart, 2002), the related software modules should be preserved in a live environment and named explicitly in the written documentation. The internal communication process- that is the interlinking of different script modules - has to be documented and may be supplemented with for example an UML-diagram.

Examples II

Before summarizing standard requirements for art-server, two examples are given.

In order to demonstrate the dependence on pre-installed server libraries Cornelia Sollfranks artwork *Net.Art Generator No. 5* (1999-2005) can be mentioned. The artwork presents an image-forming engine which enables users to generate their own art-picture out of web content (Figure 6). As the artist states: “The *Net.Art Generator* automatically produces net.art on demand” (Sollfrank, 2004). The user is invited to enter a keyword or term in a specific search field and choose additional parameters like the number of images (two to eight images), maximal width (400 px to 1000 px) and the image format (GIF, JPEG, PNG). The NAG5-image-engine then reverts to ‘Google Images’, which provides reference links to image materials from the web.

26 effects and 13 operations from the huge software library *ImageMagick* are referred to by the so called imageagent (script), enabling the image production. This results in the appearance of the artwork relying on those *ImageMagick* operations. Whereas the current server offers *ImageMagick* version 6.3.9-9 (2008) the original one was version 6.2.5. (2003).

Another problem can be illustrated with Olia Lialinas artwork *My Boyfriend came back from the War* (1996). The user can navigate through a mostly pre-designed dialog between a soldier and his girlfriend. Olia Lialina has pointed out that the intervals between the user’s click and the feedback appearing on the screen have continually shortened over time, but just those moments of waiting used to lend a narrative quality and a poetic tension.

A similar difference is measurable in the ASCII-version of *Deep Throat* (1972), which was translated into ASCII by the ASCII Art Ensemble in 1989. Whereas the full feature length originally took about one hour the ASCII-version today runs only twenty to thirty minutes.

Timing issues can easily be regulated by non-invasive operations – like the configuration of the server port.

Nevertheless occurring modifications like e.g. the improvement of connectivity (velocity / delivering time), changes in cross-referencing or within the directory have to be documented. Failures have to be analyzed and resolved as smoothly and as non-invasively as possible.

Standard Requirements for Art-Server

- All applications that provide services are to be named and mapped. Artistic software and data, as well as related files and environmental elements have to be localized in the directory. In addition the format has to be determined, in which the data is delivered.
- In order to sustain an authentic system setting the referenced software-libraries and the applied scripting interfaces have to be identified and documented. Core functionalities and the communication format should to be listed. For example for detecting the http-request a proxy might be used.
- Velocity is an additional subject important to the accurate rendition of an artwork. Ancient artworks frequently lack specific configurations that define the frame-rate or the delivery time of content. Due to a mixture of reasons like the improvement of Internet connections, increasing bandwidth, more powerful processors and better graphic cards, the timing of those artworks can change significantly.
- Since the emergence of meta-search engines like Google and additional services many network-based artworks include communication processes with network-based web- and news-service or even community portals. They are context-sensitive. Whereas self-contained artworks provide the required content by their own, context-sensitive ones rely on an Internet environment. By request they gather data or metadata from the Internet.

It is quite obvious that the external environment is hard to control. Nevertheless the behavior, format of communication and delivered data can be analyzed and documented. This aspect leads to the last segment, the network communication.

For the long-term preservation of the artwork it is important to be aware of the artist's choice of and involvement with specific software solutions. Useful questions might be: Why did the artist work with this software? Was the chosen software state of the art or out-dated? Has the currency of a software solution any meaning to the artist? How do artistic effect and process of development of the artworks' software relate?

Last but not least the historic dimension of the applied software should be traced – especially in a time where any surface effect is producible in different ways. The historic meaning can be related to the nature of the artwork as document and track the aging processes in an art-technological sense. Aging processes are part of the artwork and should not be concealed.

Analyzing and Understanding the Network Communication

Concerning network communication we want to know especially who is communicating with whom? What is communicated? How does the communication take place?

This information is essential - especially when the communication process is interrupted or modified and has to be patched.

To illustrate this, two examples of distributed communication processes shall be given. In both cases the client does not only receive data from the art-server but engages additional services from external data providers.

Examples III

First the *OneWordMovie* (2003) by Beat Brogle and Philipp Zimmermann can be mentioned. The user is invited to enter a term or keyword in a dedicated search field. As the title

suggests, a movie is generated out of the request related to the keyword. In addition the user can decide on certain aspects like frame-rate (per second), choose a number of repetitions and the sequence of collated images. The image material is derived at from the Internet. Display and overall design are realized in a Director application. The search request is implemented on three distributed server engines. They balance the load and buffer the request. The client subsequently downloads the referred images. The art-generating process is mostly automated and processed on the client.

The second example is Marc Lee's installation *Breaking the News. Be a News-Jockey* (see above). It does not only access one but 16 different web-services. Whereas the *OneWordMovie* is meant to be used on a personal computer, *Breaking the News* is a four screen projection which covers the front of a huge installation box (about 4.8m x 2.7m). The screen is never empty. The user can participate by entering keywords in a specific search field on the screen. As a result the piece delivers heterogeneous audio-visual content from the Internet and presents it in a kind of interactive, real-time wallpaper. As already mentioned earlier, the client then completes several action-script commands that are embedded in a Flash-movie.

The internal communication process of *Breaking the News* is similar in structure to *OneWordMovie*. The client delivers search requests to the art-server. In *Breaking the News* the request is then sent through a set of shell-scripts assembled by the artist. The art-server forwards the request to selected web-services and finally responds by delivering a XML-string to the client. This XML-string contains hyperlinks to web content that is continuously downloaded and displayed by the client.

Tracing Network Communication

A tool capable of tracing such complex communication processes is the so called Netart Router. The Netart Router is adapted to the specific requirements of analyzing and documenting net art and provides different functionalities. It can be used as an analysis tool and as (transparent) proxy. The position within the communication infrastructure (network) can vary depending on interest. The Netart Router may observe the communication between client and art-server or between the art-server and the external environment of the Internet. The automatically generated documentation correlates with common display tools and can thus be shown in current browsers. As an analysis tool the Netart Router focuses on the http-protocol. Based on transparent proxy technologies it can trace data requests, the communication with reliable external web-services and the communication between varying sources and targets. The Netart Router can store data as well as the protocol metadata (http-request- and -reply-headers). It is capable of generating detailed reports of both single requests and request sessions, which contain detailed navigation through an artwork. Further documentation allows continuation of the observed functionalities and supports the analysis and reconstruction of the artwork.

In addition certain routing features have been implemented. They help bridge gaps within the communication process. Future incompatibilities in the network communication are one of the serious threats to net art. Quite a number of factors cause failure of network-based artworks. Web-services or content providers move, change the access rules or conditions for request, or modify their communication format.

Examples IV

Returning to reflections on preserving environmental aspects, we have two artistic examples. They demonstrate how changes in the technological environment may cause solvable or irreparable breakdown.

The first example is John Klimas' *The Great Game* (2001/2002). Originally the artwork referenced specific maps of Afghanistan during the war. Military operations of the real world were to be reflected in a sarcastic manner within the art game. Ever since the data source for these maps disappeared the interactive usage of the artwork is disrupted. The artwork's website is its own memorial.

Exactly this aspect seems to be expected by the artist, however. He might have fixed the bug in different ways. But instead of eliminating the dysfunctional part e.g. by exchanging the data source, he provided a new version of the artwork when Iraq appeared as another trouble spot. It is the subject of *The Great Game: Iraq Expansion Pack & Champagne Maker* (2003).

The second sort of failure can be observed at Cornelia Sollfranks *Net.Art Generator No. 5* (2004) which was described earlier. Here the crucial point is not that the data source disappears but the reliance on a specific communication format. The image generating engine requires data to be delivered in HTML. Since Google changed the format of its meta-information response from HTML to JavaScript the artwork is interrupted. The *Net.Art Generator*-script is waiting for the correct protocol and format. It is not able to decode the provided JavaScript content. A non-invasive operation to solve this might be the use of a proxy to mediate the communication process. In one of our case studies the Netart Router delivers a request header to Google, which pretends the using of the browser Lynx, as this still requires responses in HTML. By linking the

artwork with non-invasive tools, the historically interesting *media brake* – the shift from plain HTML to dynamic HTML with JavaScript in 2006/07 - can be compensated.

Conclusion

The following excerpts of an evaluation Schema summarize the previous considerations. They characterize the tracked modules according to three categories: *work-relevant* components, *environmental* and *not directly accessed* elements. These three categories are applied to the client-side of the artwork, the art-server and the communication process and should be described in the written documentation.

Table 1 enables charting the client activity. We differentiate between browser-based and application-based rendition. In addition a combination is possible – like browser-embedded applications. Afterwards we focus on the clients' activity in relation to the rendering mode (Figure 4.)

In regard to the art-server the applied services (Table 2), the location of the artistic data (file system) and the artists' scripts are to be mapped. While Table 3 lists all modules and software packages including version number and date, Table 4 describes all files, modules and content which have to be saved during the backup process. UML or other structuring diagrams are recommended in order to sketch the described operation logic. The higher the rendering potential is on the server the heavier a specific system environment is relied on.

Another aspect focuses on communication processes. A so called *Communication Matrix* (Table 5) depicts permanent communication partners and enables the distinction between self-contained and context-sensitive artworks. It gives a very rough overview on who is communicating with whom and enables specifying the protagonists.

The suggested mode of operation and the evaluation chart facilitate a quick approach to conserving network-based artworks from an art-technological point of view. They offer checklist-like characterizations of central technical aspects and core elements of the artwork. They reveal relations between the operational and artistic work logic and point towards modules, which need to be preserved long term.

Both the tables of the evaluation chart as well as the presented procedure do not replace former documentation formats e.g. of preservation theory (AIC, 1994), but rather offer an easy and manageable toolbox for reducing complexity: describing client-side activity, server-side operations and the communication interchange. This eases the pin-pointing of *the artwork* within the distributed configuration. Foreseeable problems and failure are identified.

In addition to the chart the Netart Router was mentioned. It is a tool for very detailed forms of documentation of net art. A more elaborate paper on that subject is in preparation. The Netart Router enables the sustainable handling of Internet-based artworks and making provisions for future threats and drop-outs.

Finally we want to touch on some questions which have been arising lately regarding the aging processes of net art. They might open a wider field of discussion in the art-technological scope: Which status has a net-based artwork in terms of historic source (material)? What kind of meaning convey earlier versions of the artwork which document a certain time period? How can categories like Riegl's terms of historic value (*historischer Wert*) or value of aging (*Alterswert*) be applied to this type of art?

References

- AIC - American Institute for Conservation of Historic & Artistic Works (1994), Guidelines for Practice, in: <http://aic.stanford.edu/pubs/ethics.html> (URL consulted, November 2008).
- AktiveArchive (2006), <http://www.aktivearchive.ch/content/projekte.php/> (URL consulted, November 14th 2008).
- Brogle, B. & Zimmermann, P. (2003), *OneWordMovie*, in: <http://www.onewordmovie.ch/> (URL consulted, November 14th 2008).
- Browser History: <http://browsers.evolt.org/>, or in addition:
http://upload.wikimedia.org/wikipedia/commons/7/74/Timeline_of_web_browsers.svg
 (February 14th 2007).
- Dittmer, P. (1992-2005), Schalten und Walten. *Die Amme*, in: <http://www.dieamme.de/> (March 20th 2008).
- Enge, J. & Vasulka, W. (2003/04), *Eigenwelt der Apparatewelt. Die Netzedition*, in:
http://on1.zkm.de/zkm/meisterwerke/enge_vasulka (URL consulted, December 14th 2008).
- Granger, S (2001), “Digital Preservation & Emulation: from theory to practice”, in: ICHIM, Paris, <http://www.leeds.ac.uk/cedars/pubconf/papers/ichim01SG.html> (URL consulted, February 15th 2008).
- Greenwood, S. (1996), *Woven Present*, in: <http://www.telesthetic.org/woven/> (URL consulted, August 15th 2008).
- Hanhardt, J. (2003), “The Challenge of Variable Media”, in: Permanence Through Change: The Variable media Approach is published by the Solomon R. Guggenheim Museum, in New

- York, and the Daniel Langlois Foundation, in Montreal. The texts are in English and in French, p. 7-9.
- Hunter, J. & Choudhury, S. (2003), Implementing Preservation Strategies for Complex Multimedia Objects, Brisbane, in: http://metadata.net/panic/Papers/ECDL2003_paper.pdf (URL consulted, March 15th 2008).
- Klima, J. (2001), *The Great Game*, in: <http://www.cityarts.com/greatgame/> (URL consulted, March 12th 2008).
- Klima, J. (2003), *TGG – Iraq Expansion Pack and Campaign Maker v1.0*, in: <http://www.cityarts.com/iraq/index.html> (URL consulted, March 12th 2008).
- Laforet, A (2009), *La Conservation Du Net Art Au Musée. Les Stratégies a L’Oeuvre*, Marseille / Avignon. Web-Version: <http://www.sakasama.net/conservationnetart/these/theseannelaforet0309.pdf> (URL consulted, June 10th 2009).
- Laurenson, P. (2006), Authenticity, Change and Loss in the Conservation of Time-Based Media Installations, in: <http://www.tate.org.uk/research/tateresearch/tatepapers/06autumn/laurenson.htm> (URL consulted, October 10th 2008).
- Lee, K. et al. (2002), „The State of the Art and Practice in Digital Preservation“, in: *Journal of Research of the National Institute of Standards and Technology*, Volume 107, Number 1, January–February, p. 93-106.
- Lee, M. (2006), *Breaking the News. Be a News-Jockey*, <http://www.1ho1.net/index.php/Main/News-Jockey> (URL consulted, December 15th 2008).

Lialina, O. (1996), *My Boyfriend Came Back From the War*, in: <http://www.teleportacia.org/war/>
(URL consulted, November 20th 2008).

Lialina, O. (2008), *My Boyfriend Came Back From the War*, in:
<http://myboyfriendcamebackfromthewar.ru/> (URL consulted, November 20th 2008).

Lurk, T. (2008), "Virtualisation as preservation measure", in: *Archiving 2008, Final Program and Proceedings*, Ed.: R. Gschwind, Basel, p. 221-225.

Marinescu, D. & Kröger, R. (2007), *State of the art in autonomic computing and virtualization*, Distributed Systems Lab, Wiesbaden University of Applied Sciences, in:
<http://wwwvs.informatik.fh-wiesbaden.de/downloads/extern/pubs/techreports/STAR.pdf>
(URL consulted, October 10th 2008).

Rothenberg, J. & Bikson, T. (1999), *Carrying Authentic, Understandable and Usable Digital Records Through Time. Report to the Dutch National Archives and Ministry of the Interior.*

Rothenberg, J. (2006) *Reviewing the Erl King*, in:
<http://www.bampfa.berkeley.edu/about/ErlKingReport.pdf> (URL consulted, October 2nd 2008).

Saaze, V. van (2009). *Authenticity in practice. An empirical study into the conservation of One Candle (1988) by Nam June Paik*. In Hermens, E. & Fiske, T. (eds.) *Art, Conservation, and Authenticities/ Material, Concept, Context*. Archetype Publications: London, pp. 190-198.

Sollfrank, C. (1999-2004), *Net.Art Generator No. 5*, in: <http://nag.iap.de/> (URL consulted, June 15th 2008).

Storz, R. (1999), *The Ram Show*, in: <http://www.xcult.org/ateliers/ramshow/index.html> (URL consulted, October 15th 2008).

Tholen, G. (2002), *Die Zäsur der Medien. Kulturphilosophische Konturen*, Frankfurt.

Wikipedia contributors, 'Client (computing)', Wikipedia, The Free Encyclopedia (URL consulted, November 18th 2008).

Tables

Table 1
Client Specification

	Yes	No	Description
Client Type			
Browser based client			
Embedded application			
Application based client			
Navigation Devices (Input)			
Mouse			
Keyboard			
Special device			
Display (Output)			
Screen (analog/digital)			
TV monitor			
Beamer			
Special device			
Display Rendering			
Statically rendered			
Dynamically rendered			
Internet Connection			
Direct connection			
Bridged connection			
Source of External Data			
Data deriving from art-server			
Data deriving from external web- or news-service			
Data deriving from various external data providers			

Table 3
Backup Chart

Server Name

NAG No. 5

Module	Package	Files	File	Directory
Apache 1.2	X			
			*.conf	/etc/apache/
NAG 5		X		
			*	/data/nag
			*	/var/www/nag
			*	/usr/local/nag

Table 4
Dependency Chart


Server Name NAG No. 5

Module	Server	Application	Script	Other	Dependency
NAG 5			x		Apache 1.2
NAG 5			x		PHP 4.1
NAG 5			x		ImageMagick
Apache 1.2	x				
Perl				X	curl
ImageMagick		x			libpng

Figures

Figure 1

Reinhard Storz: *The RAM Show* (Screenshot)



The screenshot shows a website for 'The RAM Show'. At the top left, there is a navigation link '← RAM'. Below it is a circular profile picture of a man in a patterned shirt. To the right of the picture is the title 'The RAM Show' in a stylized font. Below the title, there is a horizontal line. Underneath the line, there is a paragraph of text: 'The RAM Foundation has taken up as its task the rehabilitation of amnesia-victims. Our search oriented service on the Internet attempts to spark off research on the background and on the identity of our amnesia-victims, with the help of agents and the international public. We will gratefully accept all tips as to the identity of the persons in question.' Below this paragraph is another paragraph: 'Welfare undertakings like the RAM Foundation have to find private funding today, and assert themselves in the strongly competitive market for donations. The Internet offers us two important opportunities. First of all, it enables us to reach numerous people across the globe, and prod them to co-operate with us in our research operations. Secondly, the empathy of the online-community awakened by the fate of our patients helps in consolidating the financial basis of our work. We need to thrash out all the aspects of infotainment and docu soaps in order to attain our statutory and economic goals. We do it for the cause of the rehabilitation of amnesia-patients, and do it with joy. Therefore we have entered into a contract for cooperation with «crosscomm», an agency which has specialised in Art & Show, for developing the RAMSHOW.' On the right side of the page, there is a vertical navigation menu with three items: 'Home', 'Contributions', and 'Information', each preceded by a small circular icon.

← RAM

The
RAM
Show


The RAM Foundation has taken up as its task the rehabilitation of amnesia-victims.

Our search oriented service on the Internet attempts to spark off research on the background and on the identity of our amnesia-victims, with the help of agents and the international public. We will gratefully accept all tips as to the identity of the persons in question.

Welfare undertakings like the RAM Foundation have to find private funding today, and assert themselves in the strongly competitive market for donations. The Internet offers us two important opportunities. First of all, it enables us to reach numerous people across the globe, and prod them to co-operate with us in our research operations. Secondly, the empathy of the online-community awakened by the fate of our patients helps in consolidating the financial basis of our work. We need to thrash out all the aspects of infotainment and docu soaps in order to attain our statutory and economic goals. We do it for the cause of the rehabilitation of amnesia-patients, and do it with joy. Therefore we have entered into a contract for cooperation with «crosscomm», an agency which has specialised in Art & Show, for developing the RAMSHOW.

 **Home**

 **Contributions**

 **Agents**

 **Information**

Figure 2

Concept of digital Environment. Schema for modularization.

Work – Environment - Relation

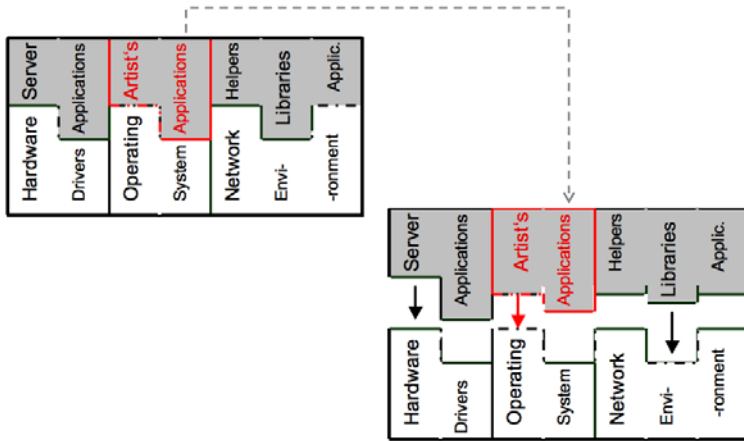


Figure 3

Standard System Architecture

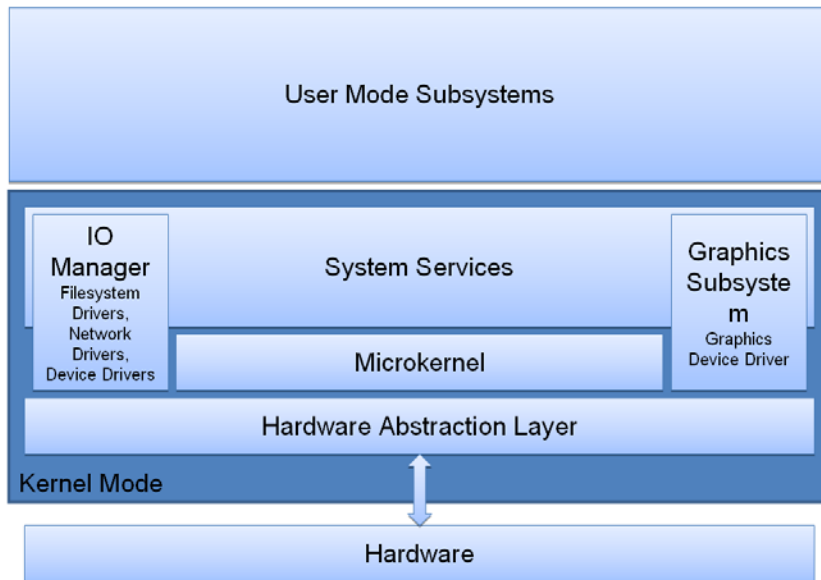


Figure 4

Graph for Localizing the Operational Logic of the Client

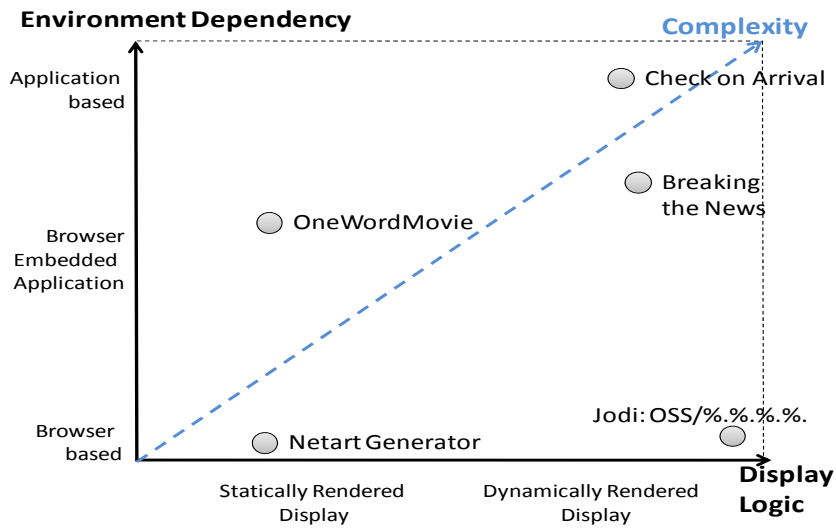












Figure 6

Cornelia Sollfrank: *Net.Art Generator No. 5* (Screenshot, Archive)

NAG :: Net.Art Generator			
Home Create Stats Deutsch			
Picture Base Status			
Net Cached: 675 (49.87 Mb) :: Generator: 4125 (499.78 Mb)			
Statistics Top 10			
	1 ich_bin-dort © Nov 29 23:41:11 9 click(s) First click: Sat Nov 29 23:41:29 2008 last: Sat Nov 29 23:42:23 2008		2 ich_bin-hier © Nov 29 23:40:51 9 click(s) First click: Sat Nov 29 23:41:24 2008 last: Sat Nov 29 23:42:20 2008
	3 schwar-17 © Nov 29 23:13:59 9 click(s) First click: Sat Nov 29 23:22:51 2008 last: Sat Nov 29 23:35:44 2008		4 anonymas-inna © Nov 24 09:04:22 9 click(s) First click: Mon Nov 24 09:07:24 2008 last: Mon Nov 24 09:10:18 2008
	5 anonymas-inna © Nov 24 09:05:00 9 click(s) First click: Mon Nov 24 09:07:14 2008 last: Mon Nov 24 09:10:10 2008		6 sWiZer-Network_Map © Nov 21 21:23:52 9 click(s) First click: Fri Nov 21 21:24:12 2008 last: Fri Nov 21 21:24:59 2008
	7 anonymous-PIPILOTTI © Nov 20 00:49:52 9 click(s) First click: Thu Nov 20 00:51:24 2008 last: Thu Nov 20 00:52:34 2008		8 suprak-mutt_1917 © Nov 20 18:54:48 9 click(s) First click: Thu Nov 20 18:56:40 2008 last: Thu Nov 20 18:57:34 2008
	9 suprak-karl-heinz_stockhausen_wtc_art © Nov 20 18:56:25 9 click(s) First click: Thu Nov 20 18:56:44 2008 last: Thu Nov 20 18:57:38 2008		10 sWiZer-I_know_all_pixels © Nov 21 21:37:05 9 click(s) First click: Fri Nov 21 21:37:33 2008 last: Fri Nov 21 21:38:16 2008