Journal of the Institute of Conservation

Routledge
Taylor & Francis Group

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/rcon20

Responding to obsolescence in Flash-based net art: a case study on migrating Sinae Kim's *Genesis*

Anna Mladentseva

To cite this article: Anna Mladentseva (2022) Responding to obsolescence in Flash-based net art: a case study on migrating Sinae Kim's *Genesis*, Journal of the Institute of Conservation, 45:1, 52-68, DOI: 10.1080/19455224.2021.2007412

To link to this article: https://doi.org/10.1080/19455224.2021.2007412

Journal of the Institute of Conservation, 2022
Vol. 45, No. 1, 52–68, https://doi.org/10.1080/19455224.2021.2007412

Anna Mladentseva
University College London, Gower Street, London WC1E 6BT, UK

Responding to obsolescence in Flash-based net art: a case study on migrating Sinae Kim's *Genesis*

Abstract
Many internet artworks from the mid-1990s to the early 2000s used Adobe Flash technology for creating animated content. However, in the light of recent web standard developments (HTML5), Adobe has stopped supporting Flash and its related tools. The removal of Flash has made those net artworks non-functional and unviewable, including Sinae Kim's *Genesis* (2001), the focus of this study. Recently proposed emulation- and virtualisation-based strategies are not always suitable, particularly if there is a desire to keep the artwork on the 'live web'. This article outlines an alternative method of migration facilitated by reverse engineering techniques—specifically decompilation—and foregrounds the significance of maintaining online access to the obsolete Adobe Shockwave Flash (SWF) files through the source code. On this premise, the source code is re-imagined as a site for further re-enactment, allowing a departure from its current role as a marker of 'authenticity'.

Keywords
internet art; Adobe Flash; time-based media conservation; source code; re-enactment; reverse engineering

Introduction

As of 31 December 2020, Adobe discontinued its Flash plug-in for the web, leaving many internet artworks that rely on the Flash player non-functional and unviewable.[1] Adobe stopped supporting Flash on the basis that HTML5—the most recent iteration of the mark-up language used for webpages—integrates multimedia content without the need for additional plug-ins and is therefore less prone to security exploits.[2] Amongst these non-functioning net artworks is Sinae Kim's *Genesis* (2001) (Fig. 1) currently hosted on Rhizome's ArtBase—an online repository dedicated to displaying and preserving new media art.[3] *Genesis* is a website combining HTML, CSS and JavaScript technologies together with embedded Flash animation files. These files have the now obsolete SWF (Adobe Shockwave Flash) extension which would have been executed by the browser's Adobe Flash player.

In *Genesis*, these Flash animations are called up using pop-up windows in response to the user clicking through a menu of options. According to the artist, *Genesis* employs references to Judeo-Christian beliefs on the creation of the world to illustrate the possibilities of the early web.[4] Following the splash page, which is titled with the opening phrase of *The Book of Genesis*, the user is presented with a black homepage that has a menu of elements on its left-hand side. Depending on which element is clicked by the user, the pop-up windows change colour, vibrate and arrange themselves in ways that imitate the selected entity. For example, the 'sun and moon' option generates two sets of windows: on the left-hand side of the page there are four red windows representing the sun meanwhile on the other side a more elongated, yellow coloured window represents the moon (Fig. 2). While this option doesn't contain any Flash animations, other options such as 'tower of babel' do.

1 In addition to the development of better performing and more secure open web standards such as HTML5, WebGL and WebAssembly, browser vendors including Apple, Google, Microsoft and Mozilla have been systematically discouraging and phasing out Flash by keeping their packages off by default, thereby contributing to its eventual obsolescence. For the original announcement, see https://blog.adobe.com/en/publish/2017/07/25/adobe-flash-update.html#gs.6c24bj (accessed 20 July 2021).

2 For Adobe's Flash Player's end-of-life information page, see https://www.adobe.com/uk/products/flashplayer/end-of-life.html (accessed 20 July 2021).

3 For Kim's work on ArtBase, see https://artbase.rhizome.org/wiki/Q2082 (accessed 20 July 2021).

4 Sinae Kim, personal communication, 27 February 2021.

Contact: Anna Mladentseva (anna.mladentseva.18@ucl.ac.uk)
(Received 2 June 2021; Accepted 12 November 2021)

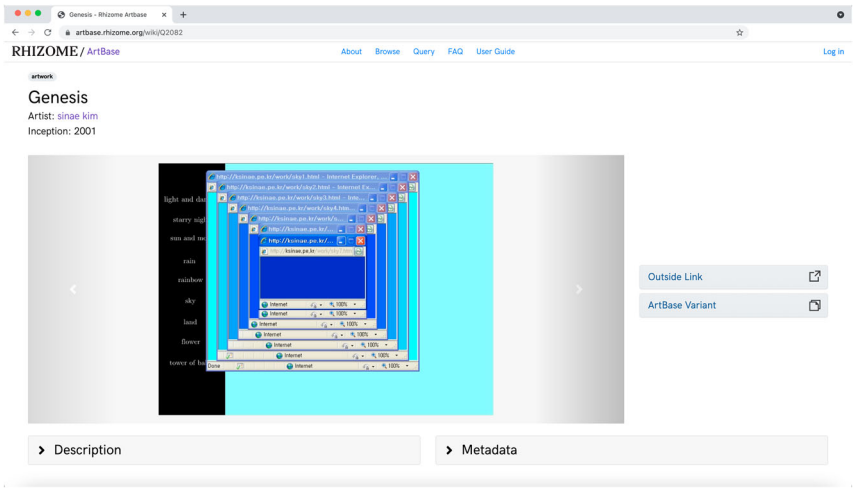

Fig. 1 Sinae Kim's *Genesis* (2001) as featured on Rhizome's ArtBase.

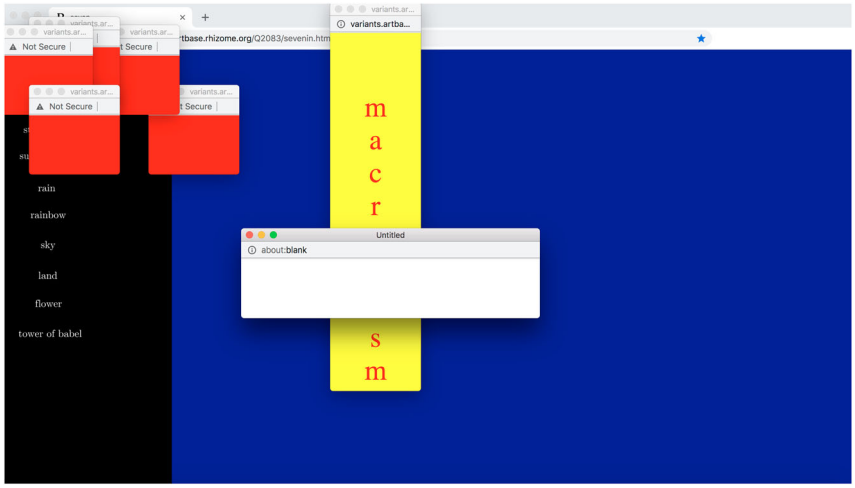

Fig. 2 'Sun and moon' animation from Sinae Kim's *Genesis* (2001) as rendered in Rhizome's ArtBase version.

Flash player's end-of-life poses a series of questions and challenges for Flash-based net art. This includes considering the methods available to artists and institutions for restoring these works, as well as the effects these methods have on the original source code. Using *Genesis* as a case study, this article attempts to begin answering some of these questions by situating first-hand empirical data and an artist interview within the wider debate on restoring Flash-based net art. Moreover, the role of the source code will be explored within the dynamics of two conservation narratives that seem to exist in tension with each other. One approach classifies network-based art as immaterial and thus ontologically similar to performance; while the other, proposed by researchers at PAMAL (Preservation & Art–Media Archaeology Lab) in Avignon, attempts to reinsert these works into a more 'material' rhetoric that foregrounds intervention into their 'lowest conditions'.[5]

A large number of institutions have been researching the potential of emulation- and virtualisation-based strategies for the conservation of time-based media.[6] These solutions are capable of providing an interface for accessing Flash-based net art by encapsulating artworks in an environment where the plug-in is still supported. In fact, Rhizome has

5 Emmanuel Guez et al., 'The Afterlives of Network-based Artworks', *Journal of the Institute of Conservation* 40 (2017): 105–20 (see p. 112).

6 This study will refer to emulation- and virtualisation-based strategies at the

Tate (London), LIMA (Amsterdam) and Rhizome (New York).

publicly announced that its response to the 'Flash sunset' has been to invest in improving its Emulation-as-a-Service (EaaS) infrastructure: a cloud-based framework for managing multiple emulated or virtualised environments, which makes them accessible through a web browser.[7]

The availability of web archiving tools that utilise these legacy environments through remote browsers—such as Rhizome's recently rebranded web archiving tool Conifer—have made access to Flash-based artworks possible. However, it is important to note that in web archiving, the resulting object is a replayable web archive (stored on a WARC file) rather than an emulator, despite its use of emulation/virtualisation during the process of 'capturing' the webpage. Notwithstanding, artists have been using Conifer to create archived versions of their Flash-based net art in order to later link them at the original non-functioning site.[8] Evidently, besides Emulation-as-a-Service initiatives, web archiving tools have also become a strategy for remedying the obsolescence of Flash-based net art and thus also deserve attention.

While these strategies are productive in providing an infrastructure that is able to preserve many artworks at once with minimal labour, the purpose of this article is to offer an alternative method of migration aimed to assist artists who prefer to maintain their work on the 'live web'.[9] This method may also be suitable for individuals who do not have access to emulation- and virtualisation-based tools. The migration process is facilitated by the capacities of reverse engineering and decompiling the obsolete SWF files for further interrogation in Adobe Animate.[10] The components derived from this interrogation accelerate the migration process because the original vector graphics that made up the SWF file may be reused by the conservator. Additionally, decompilation recovers the scripts used by the artist in the creation of the animations. These scripts are normally written in ActionScript, the programming language developed exclusively for Flash applications, which is semantically similar to JavaScript, one of the languages forming the HTML5 standard. In this way, Animate provides the conservator with an integrated development environment where the scripts and graphical components can be reworked into the updated web standard.

The proposed method becomes an invitation to negotiate the tension between the 'de-materialising' and 're-materialising' tendencies circulating in time-based media conservation. The de-materialising tendency appears to displace the 'score' of the work, which could be read as the source code, in favour of its activation.[11] In this way, the work's activation and future conservation is inextricably tied to its playback, rather than to the preservation of the 'carrier' that makes it possible. The re-materialising tendency expresses fragility in this playback, which manifests through slips in the work's appearance and sometimes in the disappearance of content. Taking this into account, the re-materialisation of an object proposes a return to the conservation of the layers that form its material conditions, particularly in cases where the media is 'dead' and the technologies enabling its playback are obsolete.[12] Despite their differences, the two tendencies, one performative and the other media-archaeological, converge in the pursuit of difference. This difference implies that the restored object is not a slavish reproduction but instead what PAMAL calls a 'second original'.[13] On this premise, it will be argued that the source code becomes a site for the artwork's re-enactment, where online access to the obsolete components is maintained: a conception that opens up new possibilities for the source code beyond its role as a score that sanctions authentic instantiations.

7 See, Dragan Espenschied, 'Emulation or it Didn't Happen', Rhizome, https://rhizome.org/editorial/2020/dec/21/flash-preservation/ (accessed 28 February 2021).

8 See, for example, https://bram.org/karaoke/karaoke.htm (accessed 31 May 2021).

9 In the context of this study, migration is defined as upgrading a medium to its contemporary standard. For more on this definition, see Jon Ippolito, 'Accommodating the Unpredictable: The Variable Media Questionnaire', in *Permanence Through Change: The Variable Media Approach*, ed. Alain Depocas, Jon Ippolito, and Caitlin Jones (New York: The Solomon R. Guggenheim Foundation, 2003), 47–53 (see p. 51), http://web.archive.org/web/20190416182413/http://www.variablemedia.net/pdf/Ippolito.pdf (accessed 13 November 2021).

10 Adobe Animate is a vector-based animation software developed by Adobe that eventually replaced the Adobe Flash software. The initial version of Adobe Flash was developed exclusively for Flash-based multimedia applications; however, in 2015, due to the product's decreasing usage, Adobe rebranded 'Adobe Flash Professional CC' as 'Adobe Animate CC'. Animate supports both Flash-based content as well as HTML5, allowing its users to use the software for migration. For the official product page, see https://www.adobe.com/uk/products/animate.html (accessed 20 July 2021).

11 Cf. Pip Laurenson, 'Authenticity, Change and Loss in the Conservation of Time-based Media Installations', *Tate Papers* 6 (2006), http://www.tate.org.uk/research/publications/tate-papers/06/authenticity-change-and-loss-conservation-of-time-based-media-installations (accessed 31 May 2021).

12 Guez et al., 'The Afterlives of Network-based Artworks', 112.

13 Guez et al., 'The Afterlives of Network-based Artworks', 119.

An overview of emulation- and virtualisation-based strategies

Generally speaking, virtualisation is a methodology for allocating computer resources in a way that abstracts the logical object from its physical component.[14] Essentially, this makes running multiple computer environments—'sandboxed' or isolated from the host operating system—on the same hardware possible. From the point of view of the machine, isolation is essential in order to create a fully controllable environment where it, depending on whether a virtualiser or an emulator is implemented, can be re-programmed to understand instructions it no longer supports or was not designed to understand. The machine will treat these sandboxed software reproductions as if they were separate pieces of hardware. This creates the possibility of implementing legacy environments in a way that allows the user to access and interact with outdated digital artefacts through their 'original' environments. Besides, sandboxing ensures that certain virtualisation-based applications, such as web archiving, are secure. This is to prevent users from accidentally leaking sensitive or personal data through a public web archive.

In addition to remote browsers for web archiving, virtualisation laid the groundwork for Emulation-as-a-Service.[15] While we can refer to virtualisation and emulation in conjunction because of this context, it is important to note that emulation goes beyond virtualisation in that it expresses a complete break in dependency from the underlying hardware. The main difference between emulators and virtualisers is that the former are capable of holding applications on platforms that are fairly distinct from the standard, contemporary computer architecture (that is, Intel-based x86 architecture).[16] For this reason, Emulation-as-a-Service—which refers specifically to the University of Freiburg's initiative bwFLA—is not necessarily limited to net art and can be employed in other software-based artworks that have been built on severely outdated or obscure hardware.[17]

Back in 2014, Rhizome's Preservation Director, Dragan Espenschied, implemented Emulation-as-a-Service to provide access to Cory Archangel's *Bomb Iraq* (2005), a software-based artwork consisting of a homemade game on a Macintosh TV.[18] Given the peculiarity of the hardware in addition to its rarity—only 10,000 units were produced—emulation seemed to be the most suitable strategy. The on-demand web client access that Emulation-as-a-Service offers allows Archangel's work to be accessed through an isolated container on Rhizome.[19] Unfortunately, the main downside of fully emulating the underlying hardware is reduced performance.[20]

Due to such issues in performance, Tate's report on emulation- and virtualisation-based strategies recommends the use of virtualisation (sometimes referred to as 'virtual machines') rather than emulation if the target digital artefact is compatible with contemporary processor architecture.[21] This can be done using the University of Freiburg's Emulation-as-a-Service framework (bwFLA) as it supports both emulated and virtualised environments, and the use of both techniques for Emulation-as-a-Service is also reported on by researchers Claudia Roeck and Beeld en Geluid for the Dutch Digital Heritage Network.[22]

For the report, LIMA (Amsterdam) and Rhizome collaborated on implementing some browser emulations for ArtHost, a platform that LIMA offers to artists and institutions for storing complex digital objects, so as to compare the effects that a browser may have on an artwork's behaviour.[23] These 'browser emulations' were set up using Emulation-as-a-Service in order to access five case studies that contained Flash animations and

14 See, for example, Matthew Portnoy, *Virtualization Essentials* (Hoboken, NJ: Wiley, 2012), 2.

15 Cf. Dirk von Suchodoletz, Klaus Rechert, and Isgandar Valizada, 'Towards Emulation-as-a-Service: Cloud Services for Versatile Digital Object Access', *The International Journal of Digital Curation* 8 (2013): 132.

16 Cf. Klaus Rechert, Patricia Falcão, and Tom Ensom, 'Introduction to an Emulation-based Preservation Strategy for Software-based Artworks' (report for Tate's PERICLES project, 2013–2017), 14, https://www.tate.org.uk/about-us/projects/pericles/emulation-based-preservation-strategy-for-software-based-artworks (accessed 31 May 2021).

17 Cf. Suchodoletz, Rechert, and Valizada, 'Towards Emulation-as-a-Service', 141. For information about the University of Freiburg's bwFLA digital preservation approach, see Klaus Rechert et al., 'bwFLA—A Functional Approach to Digital Preservation', *PIK—Praxis der Informationsverarbeitung und Kommunikation* 35, no. 4 (2012), https://www.degruyter.com/document/doi/10.1515/pik-2012-0044/html (accessed 13 November 2021).

18 Dragan Espenschied, 'Acknowledgement, Circulation, Obscurity, System Ambience', Rhizome, https://rhizome.org/editorial/2014/jun/24/emulating-bomb-iraq-arcangel/ (accessed 31 May 2021).

19 See https://rhizome.org/editorial/2014/jun/24/emulating-bomb-iraq-arcangel/ (accessed 31 May 2021).

20 Rechert, Falcão, and Ensom, 'Introduction to an Emulation-based Preservation', 15.

Java applets (another obsolete plug-in). In practice, it is not the browser that is being emulated, but rather the hardware required to run it.[24] This meant that more recent browsers (e.g. Firefox 66) that can be run on operating systems such as Windows 7 were installed on Oracle's VirtualBox: a virtualisation platform, whereas older browsers (e.g. Firefox 3.0.5) that run on Windows XP were emulated using QEMU: an emulator.[25] Despite the application of both emulation and virtualisation, the discipline still refers to the technique as browser *emulation*, though the expression 'remote browser' has also been widely used.

Other examples of virtualisation-based environments include a variety of works that formed part of Rhizome's 'Net Art Anthology' (2016–2019): an online exhibition that sought to develop and re-tell net art history.[26] For instance, *Telepresence 2* (2002) by Corpos Informáticos is exhibited using a remote Mozilla Firefox browser (version 49.0.1).[27] This browser is encapsulated by a virtual machine rather than an emulator as it has the ability to run on recent operating systems that are compatible with contemporary processors.

Motivation

Rhizome have yet to reveal how their Emulation-as-a-Service will accommodate Flash-based art. However, given the possibilities of the emulation- and virtualisation-based strategies outlined above, artists can already opt in to use individual emulators and virtualisers (such as QEMU and VirtualBox) to set up their own legacy environments. Given the recent obsolescence of Flash player, a virtualisation-based solution should suffice. As said before, some artists have also been using Conifer to create surrogates in the form of replayable web archives of their Flash-based work.

Conifer is a website tool that makes web archiving net art accessible to a mass audience, including artists. It relies on browser emulations—similar to those implemented by LIMA in their study—to provide remote access to different versions of the browser.[28] However, the user doesn't have the means to customise these browsers to their needs, suggesting that if an artist is unhappy with the way in which the provided browsers execute their archived work, they may have to use another tool. In Conifer, the remote browser is encapsulated in a docker container and overlaid with the platform's interface. Its purpose is to allow users to record the contents of the remote browser using a capture button. As a form of symmetrical web archiving, only the contents and interactions that a user performs during the recording process will be viewable and replayable in the final web archive.[29]

Currently, Conifer lacks a centralised repository that indexes all public web archives, though Rhizome is planning on improving the discoverability of other users' captures in the future.[30] Nonetheless, individuals can make their web archive public by sharing its link with their audience with, for example, the artist Annie Abrahams linking a web-archived version of her work *Karaoke* (2006), which features Flash animations, at the original website.[31] In this way, if an internet user comes across the unviewable work at its original site, they are redirected to Abrahams' web archive where they can view the Conifer version of the website together with its Flash animation. Other ways in which these surrogate web archives have been used include Rhizome's Webenact server, where a work's replayable capture is integrated into the artist's original homepage, creating a more seamless experience.[32]

The creation of an interactive web archive using Conifer takes significantly less time than the implementation of an emulator or a virtualiser; however, it has its disadvantages. All three legacy browsers that Conifer offers execute *Genesis* incorrectly. The 'tower of babel' option, which is

21 Rechert, Falcão, and Ensom, 'Introduction to an Emulation-based Preservation', 15.

22 Claudia Roeck and Beeld en Geluid, 'Web Browser Characterisation, Emulation and Preservation' (report for Dutch Digital Heritage Network, January 2021), https://publications.beeldengeluid.nl/pub/1863 (accessed 20 July 2021).

23 Roeck and Geluid, 'Web Browser Characterisation', 6.

24 Roeck and Geluid, 'Web Browser Characterisation', 6.

25 Roeck and Geluid, 'Web Browser Characterisation', 26. The Oracle VM VirtualBox is open-source software for x86 virtualisation; QEMU is an open-source emulator; https://www.qemu.org/ (accessed 13 November 2021).

26 See https://anthology.rhizome.org/ (accessed 31 May 2021).

27 See https://sites.rhizome.org/anthology/telepresence2.html (accessed 22 May 2021).

28 Currently Conifer offers three browser emulations that are installed on a Linux operating system all with the same version of Adobe Flash Player 32.0.0.223: Chrome v76, released on 5 August 2019; Firefox v68, released on 9 July 2019; and Firefox v49, released on 23 September 2016. A fourth option is to use the user's current browser.

29 See Ilya Kreymer and Morgan McKeehan, 'Symmetrical Web Archiving with Webrecorder, a Browser-based Tool for Digital Social Memory. An Interview with Ilya Kreymer', *The National Digital Stewardship Residency*, 2016, https://ndsr.nycdigital.org/symmetrical-web-archiving-with-webrecorder-a-browser-based-tool-for-digital-social-memory-an-interview-with-ilya-kreymer/ (accessed 22 May 2021).

30 Mark Beasley (Rhizome's Lead Developer), personal communication, 12 May 2021.

31 See https://bram.org/karaoke/karaoke.htm (accessed 31 May 2021).

32 See https://webenact.rhizome.org/ (accessed 22 May 2021).

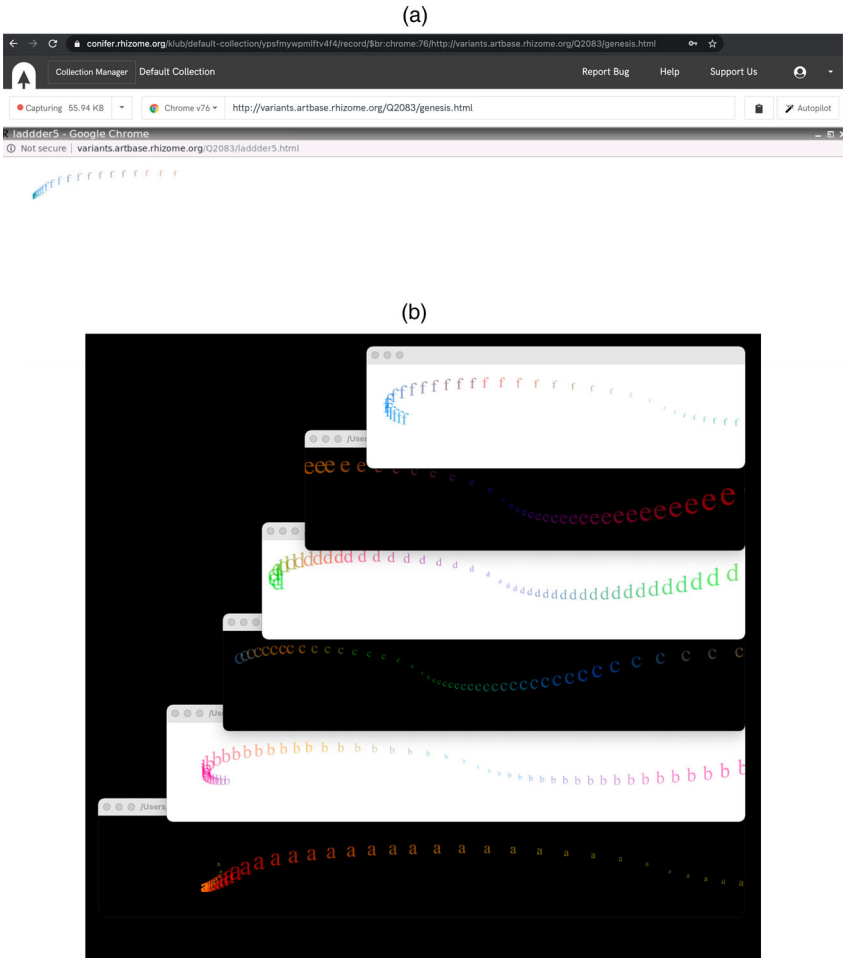

Fig. 3 (a) The remote browser in Conifer (Chrome v76) incorrectly executing the pop-up function of the 'tower of babel' animation. (b) An accurate reconstruction of the 'tower of babel' animation on the author's desktop using the original SWF files.

the last option on the left-hand side menu, consists of six rectangular pop-up windows that are stacked on top of each other in the form of a ladder. Each window has a looping Flash animation, where a rainbow-coloured letter moves in an ellipsis. All six windows spell out 'abcdef' and alternate between a black or white background. In Conifer, upon clicking the 'tower of babel' option, the Chrome v76 browser emulation disregards the set sizing of the pop-up windows and stacks them on top of each other (Fig. 3). In the Firefox v68 and v49 browser emulations, a faint outline of the correct sizing and composition can be observed for a few moments, but once the pop-up windows load their contents, they also end up stacked on top of each other. As expected, the fourth, 'current' browser option fails to load the page entirely. Given Conifer's limited capacities (in terms of customising the browser emulations), the only way to access the work properly through emulation/virtualisation is to implement a fully functioning emulator or virtual machine, though even this option is not always ideal due to personal preferences.

Primarily, Kim's work is interested in the native form of the browser and its capabilities.[33] This is certainly true not only for *Genesis*, but also Kim's other work. In *Browser Abstract* (2006), Kim superimposes multiple frames to create a maze of windows and scroll bars, reducing it to a schematic, abstract representation. It is not surprising that during our interview, Kim suggested reconstructing the animations in *Genesis* using the visual

33 Sinae Kim, personal communication.

programming language Processing.[34] Animations made in Processing can be easily deployed onto the web with the help of the JavaScript port Processing.js, which allows PDE (Processing Development Environment) files to be inserted into an HTML canvas tag, although, the recent development of p5.js, a JavaScript creative coding library which is officially supported by the Processing Foundation, has made the Processing.js technique unsupported.[35] Either way, the ability to implement these Processing-based animations onto the modern web would have allowed Kim to replace the embedded SWF files while maintaining the overall structure of the webpage. After all, it was felt that in order to stay true to the work's intention, which tells the story of the beginnings of the world and the beginnings of the web, *Genesis* had to stay on the 'live web'. Because of the challenges in executing the work through Conifer, alongside the preference of keeping the work on the 'live web', alternative methods can be made available to guide artists such as Kim through a migration path that focusses on re-activating existing material, rather than a complete reconstruction using alternative languages such as Processing.

Method

Before delving into the proposed migration path, it is important to acknowledge the conditions in which the following research has been conducted. The very nature of net art is that it resides on the world wide web—an open arena where, potentially, any user can access and inspect its networked phenomena. This accessibility proved to be particularly beneficial for my research as the contents of *Genesis*—that is, the scripts required to run it—are mostly contained on the client's side, the browser. This implies that its behaviour can be analysed without back-end access to Rhizome's server, which is responsible for hosting the website. Therefore, I was able to make use of the browser's developer tools, such as right-clicking and selecting the 'view page source' command, to read these scripts, and more importantly, retrieve the SWF files for decompilation.

The navigations in *Genesis* are contained within SWF files in the form of event listeners that are attached to specific vector graphics using ActionScript, thereby turning them into 'buttons'. Therefore, after the removal of Flash in December 2020, it was impossible to go past the splash page and navigate the website, with the grey overlay warning that 'Adobe Flash is no longer supported' prohibiting the user from right-clicking and inspecting the webpage. In these instances, I navigated the website by directly entering the desired path. For example, to access the homepage from the splash page, I changed the end of the URL from 'genesis.html' to 'sevenin.html'. The URL path can also be changed in order to request the browser to download the SWF files. Where I wasn't sure of the filename, I was able to make an educated guess. For example, the 'tower of babel' animation contains six separate Flash animations, one of which is saved under the name 'laddder.swf'. It could be deduced, by trial and error, that the other filenames were 'laddder1.swf', 'laddder2.swf', etc. In cases where only a portion of the website was blocked off by the Flash plug-in warning, such as the homepage, I navigated the site at the level of the source code—that is, by clicking on the hyperlinked elements within the code.

As such, before even starting any decompilation process, it may be useful to navigate the target website using the suggestions above and record which elements of the website need to be migrated and which don't. In the case of *Genesis*, not only did the actual animations use Flash, but also the entire splash page and the left-hand side menu from the homepage. In contrast, four out of nine options (including 'sky', 'starry night', 'sun and moon' and 'rain') use plain-coloured pop-up

34 Sinae Kim, personal communication.

35 As of 2018, Processing.js has been archived on GitHub (see https://github.com/processing-js/processing-js) due to advances in the native JavaScript creative coding library p5.js. Users can still implement Processing sketches onto the web using Processing.js, though they won't be able to apply some of the new features introduced in Processing v3. For the official page of the Processing Foundation, where you can find out more about p5.js, see https://processing.org/ (both accessed 10 August 2021).

windows with JavaScript enabling their behaviour and therefore didn't need to be migrated.

It could also be useful to execute the Flash-based animation files locally in an offline version of Adobe Flash player. This is particularly convenient in cases where documentation is unavailable. At the same time, it is important to acknowledge the security risks associated with using software that has approached its end-of-life. Adobe warns its users to de-install Flash player from their systems, as the lack of updates and patches makes it prone to security exploits,[36] although its continuing decrease in usage makes any such attack unlikely.

36 See https://www.adobe.com/products/flashplayer/end-of-life.html (accessed 31 May 2021).

1 Decompilation

Decompilation is a common practice in software engineering that extracts human-readable code from an executable stored in a machine-readable format, which in this case is the SWF file. This is possible because the machine-readable contents of the SWF file were once coded in a high-level programming language using an integrated development environment. SWF files are compiled from ActionScript, the language developed by Adobe from ECMAScript, a standard it shares with JavaScript, for Flash-based applications. These high-level programming languages are highly abstracted from machine language and are therefore human-readable.

Decompilation falls under the technique of reverse engineering, which Tom Ensom divides into two phases: decompilation and source analysis.[37] Ensom expresses a vision of reverse engineering tools as negotiating the opaqueness of processes that take place before a software-based artwork's actualisation. Indeed, decompilation becomes a means for reverting back to a point in time when these animations were being made, giving insight into the artist's working process. At the same time, Ensom notes that peeking into these 'hidden' computational processes using the above instrumentation is becoming increasingly challenging in the context of proprietary software and 'operates in a legal grey area'.[38]

37 Tom Ensom, 'Revealing Hidden Processes: Instrumentation and Reverse Engineering in the Conservation of Software-based Art', *The Electronic Media Review* 5 (2017–2018), https://resources.culturalheritage.org/emg-review/volume-5-2017-2018/ensom/ (accessed 31 May 2021).

38 Ensom, 'Revealing Hidden Processes'.

Fortunately, SWF files can be decompiled by the free and open source JPEXS Free Flash Decompiler, making it possible for us to peel away the material conditions that comprise Flash-based net art.[39] By far, the most important file that could be recovered from decompiling *Genesis* is the FLA file, which serves as an intermediary between the compiled SWF file and its separate components. JPEXS Free Flash Decompiler provides a layout where all of these components are represented in a tree-like structure, facilitating navigation between the work's material layers. The FLA file can be directly imported to Adobe Animate and will migrate with it the following components: shapes; sprites; texts; buttons; fonts; frames; and scripts.

39 See https://github.com/jindrapetrik/jpexs-decompiler (accessed 31 May 2021).

2 Migration

After importing the FLA files into Animate, the proposed migration path for *Genesis* consists of a focus in the following areas:

a) re-scripting the website's navigations, which were coded in ActionScript by attaching event listeners (Fig. 4), into JavaScript; and
b) re-scripting the movement of certain clips on the 'stage', a term Adobe uses to describe the 'canvas' for an animation, from ActionScript (Figs 5–6) to JavaScript.

Once the above migrations are completed, the FLA file can be exported into HTML5, which is made up of three files: an HTML, a JavaScript and a

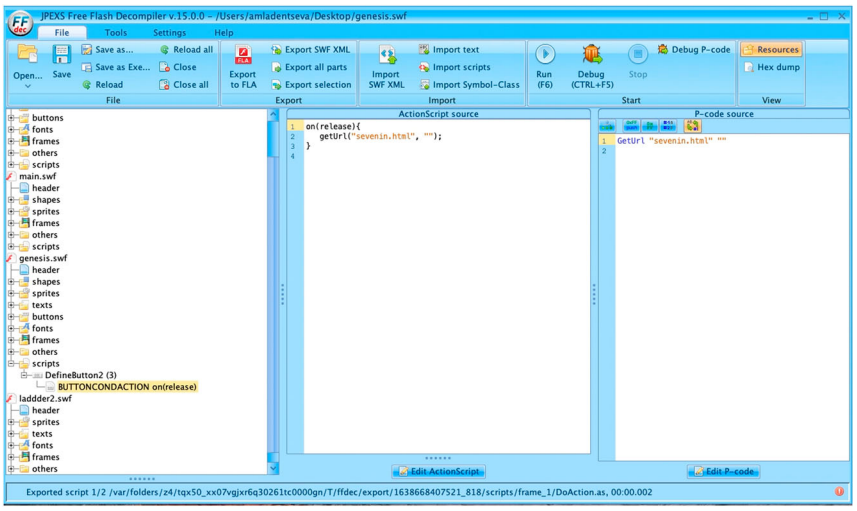

Fig. 4 Contents of the decompilation process in JPEXS Free Flash Decompiler: ActionScript for website navigations.

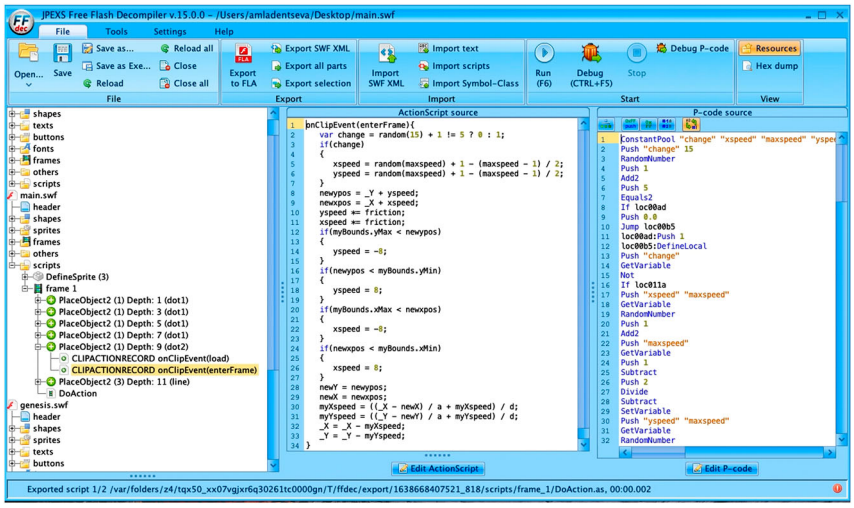

Fig. 5 Contents of the decompilation process in JPEXS Free Flash Decompiler: ActionScript for the movement of an animation in the homepage.

PNG sprite sheet file.[40] The HTML file from this package can then be integrated into the overall structure of the website through an HTML iframe tag.

The data presented in this study form part of ongoing research; however, they already point to several conclusions. Firstly, because all vector-based elements have been imported together with the FLA file (Fig. 7), time can be saved by not having to create the vector graphics from scratch. Although migration is undoubtedly laborious, the ability to reuse elements and the semantic similarity between ActionScript and JavaScript makes the process quicker. For example, the line 'stop();' inside the if statement in Fig. 6, which refers to the stopping of an animation at a given frame, has to be re-written to the JavaScript alternative 'this.stop();', while the button event 'on(release){ … }' in Fig. 4 can be replaced by 'this.x.on ('click', function(){ … }' where x is the name of the button. Given the current availability of decompilation tools and Adobe Animate software,

40 A sprite sheet can be described as a PNG file containing 'stop frames' or variations of an animation.

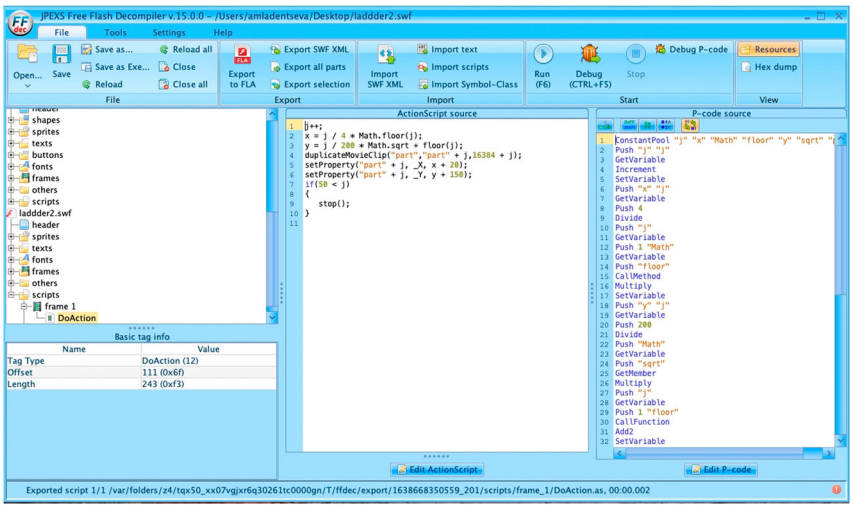

Fig. 6 Contents of the decompilation process in JPEXS Free Flash Decompiler: ActionScript for the movement of the 'tower of babel' animation.

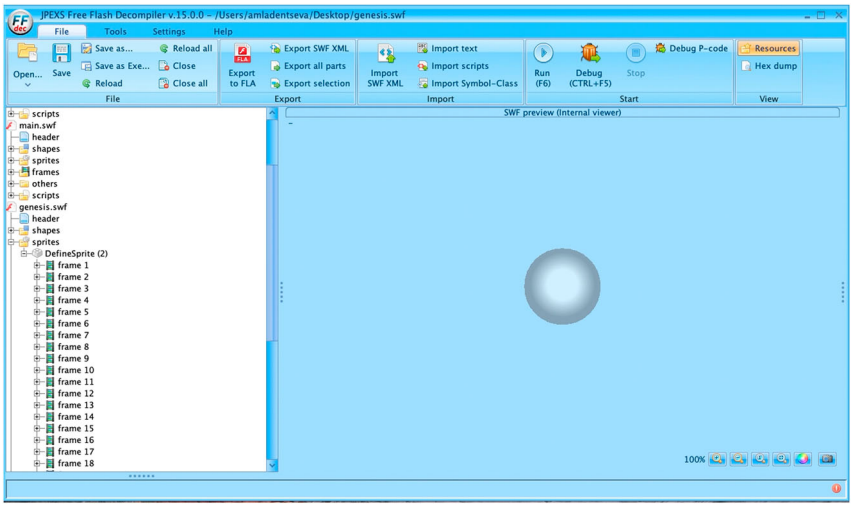

Fig. 7 Contents of the decompilation process in JPEXS Free Flash Decompiler: an example of a vector graphic (sprite) from the splash page.

this method is a suitable compromise between complete restoration as in for example, Processing, and implementing emulation- or virtualisation-based strategies.

3 Considerations on ethics, the source code and access

Re-using media assets and rewriting the artist's ActionScript code into JavaScript engages with the larger practice on source code analysis and conservation ethics in the treatment of software-based art, as outlined by researchers at the Conserving Computer-based Art Initiative at the Solomon R. Guggenheim Museum, New York. Deena Engel and Joanna Phillips argue that, 'obtaining the uncompiled, human-readable source code […] is perhaps the most crucial act of […] conservation', admitting that this sometimes requires 'decompiling the artist-provided executa-

ble'.[41] Once decompilation is performed, there are additional ethical considerations that the practitioner is advised to engage in as Phillips and Engel position the source code as a 'significant original' that needs to be respected.[42] As such, code intervention needs to be minimised even in cases where it is determined to be necessary. One way in which this can be achieved is by choosing a destination language 'similar in syntax and structure to the original', suggesting that our choice of migrating Kim's ActionScript code into JavaScript is more ethically informed than the Processing restoration alternative.[43] Another important practice that should be implemented in migrating Flash-based net art is documenting any changes made to the code by creating annotations to signal where the intervention begins.[44] This involves commenting out the original ActionScript so that it can be recorded within the human-readable iteration of the animation program, that is, in the working FLA file.

Lastly, having recovered the data outlined above, we can start to question further possibilities of the source code given that, in our case, it has become a site that sanctions further development (or recovery) of the work without back-end access to Rhizome's server. The conditions that made this particular research project possible point to the importance of maintaining the open-ended and accessible nature of net art, as it guarantees that the artwork can be intercepted and (potentially) restored by anyone on the internet.

Ensom reflects on the ways in which reverse engineering tools remain 'on the fringes of computer science' and that 'unfortunately, this process is largely hidden from a user when software is executed unless specific steps are taken to intercept or instrument it'.[45] With this Ensom is perhaps suggesting that the unclear legality surrounding the analysis of proprietary software, together with the lack of collaboration between museum conservators and reverse-engineering specialists, causes many instrumentation techniques to be left underutilised.

I would argue that we can take this concern further by thinking about the ways in which acts of time-based media conservation do not always involve collaborations with the expanded online community, which is a missed opportunity for net art given its accessible nature. The problematic of privileged stewardship in heritage has been previously examined by Emma Waterton and Laurajane Smith, who argue that the notion of the community has been institutionalised to a point where access to resources is limited unless the person in question holds the title of a 'heritage expert'.[46] This begs the following question: should information about an artwork's lifetime, including its past obsolete components, be kept within an accessible site, such as the source code, to ensure that further research is made possible? Most importantly, in line with Waterton and Smith's notion on the misrecognition of community in heritage, we have to acknowledge the possibility of research by independent conservators who may fall outside of institutions.

Unfortunately, it seems as though the isolated interaction economies present in emulation- and virtualisation-based solutions complicate net art's status as accessible, making intervention without institutional or back-end access difficult. Emulators, virtual machines and Conifer's replayable web archives are all abstracted from the user's host system, meaning that certain permissions, such as access to local storage, are usually forbidden. This makes behaviours such as downloading files from the web page difficult to perform. Moreover, Emulation-as-a-Service applications create a buffer in the form of an HTML container and its parameters, which buries the artist's source code and makes it harder to locate through the client's side. Theoretically, if a user would like to download and decompile the files comprising an artwork which is presented through an emulator or a

41 Deena Engel and Joanna Phillips, 'Applying Conservation Ethics to the Examination and Treatment of Software- and Computer-based Art', *Journal of the American Institute of Conservation* 58 (2019): 192.

42 Engel and Phillips, 'Applying Conservation Ethics', 182.

43 Engel and Phillips, 'Applying Conservation Ethics', 186.

44 Engel and Phillips, 'Applying Conservation Ethics', 188.

45 Ensom, 'Revealing Hidden Processes'.

46 Emma Waterton and Laurajane Smith, 'The Recognition and Misrecognition of Community Heritage', *International Journal of Heritage Studies* 16 (2010): 10.

virtual machine, they would have to retrieve them in officially sanctioned ways; for example, by gaining access to the relevant institution's (e.g. Rhizome's) server. Arguably, as in this project, there ought to be a consideration of how to maintain the open-ended and accessible nature of net art in a way that sanctions its further conservation with the source code as potentially the most essential actor in this.

De-materialising and re-materialising the networked object

The de-materialisation of the networked object and its comparison to performance art can be traced back to Pip Laurenson's seminal paper: 'Authenticity, Change and Loss in the Conservation of Time-based Media Installations'.[47] In her paper, Laurenson reworks Nelson Goodman's distinction between art that is autographic (paintings, sculpture) and allographic (music, theatre) to classify time-based media as allographic. At the heart of this comparison are 'the work-determinative features', or instructions, that guide a time-based media work's presentation.[48] In this way, time-based media is formed by the constant negotiation between the 'score' that specifies it and the iterations which manifest it in an installation or performance.[49] In this model, institutions often work with artists to decide the regulations under which certain components, for example playback media, are replaceable and when these items are bespoke. These so-called 'thin' and 'thick' properties determine the degree to which the work's components can be re-interpreted while also maintaining 'authenticity'.[50]

Although Laurenson's paper focusses on time-based media installations rather than networked art, software-based art conservators have been borrowing the notion of the 'score' and applying it to the source code. In their discussion of conservation ethics, Engel and Phillips argue that the 'code acts like a score that precisely reveals the rules and behaviours of an artwork'.[51] Ensom makes a similar comparison by adapting a software model from the National Archives of Australia to describe the act of restaging software performance. In this model, the technical environment—including the source code—is carefully managed to ensure 'the integrity of the [software's] performance'.[52] Previously mentioned techniques such as code annotation illustrate the ways in which this can be implemented, and Guggenheim's restoration of Shu Lea Cheang's *Brandon* (1998–) is a pertinent example of such source code management.[53] In all these cases the purpose of the source code is inextricably tied to the achievement of a faithful instantiation of a given artwork.

These de-materialisations function by creating a radical distance between the score and its iterations. Phillips expands on this distance by arguing that a time-based media artwork 'only exists when it is installed'.[54] Any material changes that occur to the work only do so periodically, that is, only when the artwork is instantiated. As a result, the performativity of the work is foregrounded and hence the claim that the object becomes 'de-materialised'. The work is pivoted around the source code (as score) which, despite being subjected to change through thinly and thickly specified properties, is quite deterministic and one-dimensional in its role of creating authenticated constraints in an otherwise 'dynamic system'.[55]

Emmanuel Guez and his colleagues in PAMAL re-materialise the networked object by emphasising the fragility of its performance. This fragility is particularly evident in so-called 'dead media', when a work has died in line with the ecosystem that once supported it. Between 2015 and 2016, PAMAL exhibited stages of their media-archaeological restoration of Eduardo Kac's *Videotext Poems* (1985–1986), which consisted of a series of primitive animations on a *Minitel*, an early example of a networked terminal designed to host a variety of online—that is, telephone

47 Laurenson, 'Authenticity, Change and Loss'.

48 Laurenson, 'Authenticity, Change and Loss'.

49 For a discussion on Laurenson's treatment of allographicity and the concept of iterations, see Joanna Phillips, 'Reporting Iterations: A Documentation Model for Time-based Media Art', *Revista de História da Arte* 4 (2015): 168–80.

50 Laurenson, 'Authenticity, Change and Loss'.

51 Engel and Phillips, 'Applying Conservation Ethics', 192.

52 Ensom, 'Revealing Hidden Processes'.

53 See http://brandon.guggenheim.org/ (accessed 20 July 2021).

54 Phillips, 'Reporting Iterations', 5.

55 Laurenson, 'Authenticity, Change and Loss'.

line—content.[56] The Brazilian iteration of the *Videotexto* system (implemented by the Companhia Telefônica de São Paulo), on which Kac's animations were stored, only functioned until the mid-1990s and as of June 2012, all servers that formed part of *Minitel*'s infrastructure in France shut down. Given the obsolescence of both infrastructures, PAMAL took on an elaborate process of reconstructing the work by navigating through several layers of its materialities. These materialities ranged from low, bit-level programming (stored in hexadecimal code) for the animations, to setting up a micro-server in order to simulate a 1980s *Minitel* server, thereby allowing anyone who might still own a *Minitel* to dial-in and connect to the reconstructed work.[57] By recreating the original material conditions as closely as possible, PAMAL arguably invert the performance-based model and its displacement of the 'score'.

PAMAL's aim to re-materialise the network-based object is closely aligned with the pursuit of the 'second original': a term they introduce to describe their work on Kac's *Videotext Poems*. According to PAMAL, 'the aim of creating a second original is not born out of the necessity to maintain an artwork as in any way authentic'.[58] In this vein, the so-called 'score' is unzipped to accommodate an afterlife to objects that may require elaborate technological intervention. Indeed, net artworks are complex digital objects whose multiple internal and external cardinalities (or dependencies) mean that they cannot be reduced to a single score or source.[59] By engaging with the different programming levels of Kac's *Videotext Poems*, PAMAL foreground the multiplicity of its 'source codes'. The purpose of these unzipped, multiple 'scores' is not to instruct an authenticated performance, but rather to express 'writing' possibilities.[60] It is well-known that there are many ways of achieving similar computational behaviours, which is why Emmanuel Guez argues that 'the notion of original writing seems to have disappeared'.[61]

PAMAL recognise the radical distance between the so-called 'score' and its iteration, which explains why they are sceptical towards applying the model of performance art onto network-based art: 'considering artworks as scores [… is] arguably based on an immaterialist conception of art where the ambition is to preserve the "soul" of the artwork, not the material "body"'.[62] To exemplify their position, the group comments on Kac's own re-interpretation of *Videotext Poems* for Whitechapel Gallery's (London) 'Electronic Superhighway' exhibition (January–May 2016), where a video reconstruction of the work was played on 2015 PC tablets and 'encased in the shells of black and white *Minitels*'.[63] Of course, this version is a 'second original' in its own right, informed by the artist's memory of the lost original, that attempts to restage an experience of the work in the present. However, this does not stop PAMAL from describing the *Minitels* as 'gutted', for none of the original electrical components or back-end infrastructure were kept intact.[64] More recently, PAMAL's Morgane Stricot, Daniel Heiss, and Matthieu Vlaminck remind us of these de-materialising tendencies, suggesting that 'these preservation methods destroy the original code/material relationship'.[65] In trying to locate the effects that the restoration of *Genesis* has on the role of the source code, we can begin to close the existing gap between the score and its iteration. This fusion creates further possibilities for the source code beyond its expression as a 'significant original', aligning it with our pursuit of maintaining the accessible and open-ended nature of net art and its conservation.

Redefining the source code as a layered site for re-enactment

At its most basic, the source code provides a set of executions that the software-based artwork follows. However, in thinking about net art, it has been

56 Guez et al., 'The Afterlives of Network-based Artworks', 112.

57 Guez et al., 'The Afterlives of Network-based Artworks', 118.

58 Guez et al., 'The Afterlives of Network-based Artworks', 119.

59 Cf. Margaret Hedstrom and Christopher A. Lee, 'Significant Properties of Digital Objects: Definitions, Applications, Implications', Proceedings of the DLM-Forum (2002): 221, https://ils.unc.edu/callee/sigprops_dlm2002.pdf (accessed 13 November 2021).

60 Guez et al., 'The Afterlives of Network-based Artworks', 111.

61 Daniel Heiss, Morgane Stricot, and Matthieu Vlaminck, 'Media Archaeological Reconstruction of Media and Digital Artworks: Practical Case Studies', in *Technological Arts Preservation* (Istanbul: Sabancı University Sakıp Sabancı Museum, 2021), 193, https://www.sakipsabancimuzesi.org/en/page/technological-arts-preservation (accessed 13 November 2021).

62 Guez et al., 'The Afterlives of Network-based Artworks', 111.

63 Guez et al., 'The Afterlives of Network-based Artworks', 119.

64 Guez et al., 'The Afterlives of Network-based Artworks', 119.

65 Heiss, Stricot, and Vlaminck, 'Media Archaeological Reconstruction', 193.

defined as both a score that holds specifications for an authentic software performance and an expression of machinic writing possibilities. Rather than stretching the source code between two slippery ontological positions which propose, respectively, a de-materialisation and a re-materialisation of the networked object, it can be formulated as an open site for re-enacting 'dead media' in ways that maintain the work's relationship with its 'lower' material conditions.

In negotiating the ephemerality of performance, re-enactments were once seen as the only worthy expression for this category of art. For Sven Lütticken, performance artists of the latter half of the twentieth century adopted re-enactments in order to create an alternative to 'the misleading representations of photography and video'.[66] The precision and accuracy of recording media is reflective of an obsession over a mythical submission to detail and exact replication, leaving no element of 'surprise' or open-endedness that exists in any *actual* performance. Moreover, performance artists and theorists from the 1960s (including Guy Debord and the Situationist International group) had a particular distrust of media, which they believed reproduced 'commodity-images', resulting in passive consumption by the viewer.[67] The rejection of playback media, which was far too symptomatic of 'the capitalist exploitation of cinema', fostered the association that re-enactments are inherently immaterial.[68]

At the same time, the purpose of re-enactment carried an additional rhetoric that sought to establish communal forms of engaging with culture or memory practice. To express this characteristic, Lütticken references the early twentieth century historical pageants staged by the English dramatist Louis Napoleon Parker.[69] These pageants combined local histories with fiction and were known to recruit the general public to perform them. As a form of participatory theatre, it collapsed 'the safe distance between performers and audience in order to create ambiguous, mixed states'.[70] Over time, the participatory nature of re-enactment has been reworked to develop conservation strategies that make use of 'delegated performances', which in turn invites us to re-imagine the term in material contexts.[71]

In this way, theories of re-enactment, though traditionally operating in the discourse of performance art, align themselves with the media-archaeological pursuit of a 'non-linear way of understanding past-presents'.[72] Gabriella Giannachi reminds us that the prefix 're-' in 're-enactment' implies both 'again' and 'back', and thus simultaneously 'a restoration and a repetition'.[73] Giannachi takes this further by arguing that the performance and documentation of an artwork that has been informed by re-enactment 'should not be looked at as a chronological progression on a linear timeline, but as a series of folds'.[74] Perhaps it is within these folds that we can collapse the radical distance between a net artwork's score and its iteration, recovering the severed relationship between a piece of code and its lower material conditions. The non-linearity of re-enactment introduces an additional dimension to this relationship, those 'writing possibilities' that manifest through past and future versions of a given artwork and its components, which overlap and form together, following Giannachi's thought, 'series of folds'.

In the context of Flash-based net art, embracing an expanded meaning of re-enactment suggests pointing to the loss and obsolescence of the SWF files while still maintaining access to them through, as suggested earlier, the source code. After all, these SWF files steer the conservator, through decompilation, towards the media-archaeological remains of discontinued ActionScript code that stands in for the work's lower conditions. As the conservator navigates through these

66 Sven Lütticken, *Life, Once More* (Rotterdam: Witte de With, 2005), 24.

67 Lütticken, *Life, Once More*, 17.

68 Lütticken, *Life, Once More*, 37.

69 Lütticken, *Life, Once More*, 31–3.

70 Lütticken, *Life, Once More*, 27.

71 For a discussion on delegating the acts of conservation, see Hélia Marçal, 'Conservation in an Era of Participation', *Journal of the Institute of Conservation* 40 (2017): 97–104.

72 Cf. Jussi Parikka, *What is Media Archaeology?* (Cambridge: Polity Press, 2012), 137.

73 Gabriella Giannachi, 'At the Edge of the "Living Present": Re-enactments and Re-interpretations as Strategies for the Preservation of Performance and New Media Art', in *Histories of Performance Documentation* (New York: Routledge, 2018), 120.

74 Giannachi, 'At the Edge of the "Living Present"', 129.

layers, they are faced with multiple 'writing possibilities' where 'dead media' (that is, the obsolete, Flash-related components) can be re-enacted, or re-activated, through a process of migration. The participatory nature of re-enactments incites these remains to be presented to the online public, integrating the obsolete components into the overall experience of the work rather than placing them out of sight. In this way, the source code operates not only as a form of version control, where these remains meticulously document and control the object's 'authenticity' in line with an established code of ethics, but also as a layered site for re-enactment where anyone on the internet is invited to access and explore it.

Changes to conservation workflows: on the importance of online access

The conditions that made the research for this study possible are inextricably linked with the practice of keeping the source code open and accessible, driven by an expanded meaning of who makes up the 'community', or perhaps even the 'conservator', in heritage. Traces of expansion and participatory involvement can be felt throughout *Genesis.* One of the HTML scripts on the site has instructions written by the artist in their native Korean language outlining, and even encouraging, the opportunity of toggling with the size of the pop-up windows. Moreover, in our interview, Sinae Kim expressed that she felt like her work could be (re)programmed by anyone.[75] Perhaps this statement can be taken both as a confirmation of the exhaustion of the source code in artistic 'originality' or authenticity and an invitation to maintain its online access in the democratic ecosystem of the internet. Having synthesised the performative and media-archaeological strands of conservation thought with the help of re-enactment, we can start to consider the ways in which this re-appraisal can unfold in practice, and how the expanded participation which the source code encourages ties in with Annet Dekker's recent contribution to the field.

75 Sinae Kim, personal communication.

In examining the future of Martine Neddam's *mouchette.org* (1996–), an expansive website featuring many user-driven contributions, Annet Dekker believes that 'a networked, community-driven conservation strategy is not unlikely to happen'.[76] Following this thought, Dekker develops the notion of 'networks of care' to describe the process of promoting a website visitor to the level of the artwork's caretaker. To illustrate this, Dekker recites an incident that happened in 2002. When Neddam was requested to take down one of the pages in *mouchette.org* due to copyright claims, several independent organisations mirrored it on their own websites.[77] Dekker's analysis points to an emergent form of conservation that involves the public in otherwise institutionally contained practices. To make this workflow possible, particular structures have to be in place that will facilitate the public's access to an artwork's components. As has been mentioned earlier, this can be achieved in Flash-based net art by storing its previous iterations, including the obsolete SWF files, in an accessible manner. One way for an institution to ensure that the expanded community can get involved when necessary would be to publish all relevant assets in a public repository, such as GitHub, under a GNU license. Alternatively, its components can be left at the work's original site in an archaeological gesture which, in line with PAMAL's thesis, allows us to consider its 'temporal and epistemological ruptures' in ways that enrich the work's meaning.[78]

76 Annet Dekker, *Collecting and Conserving Net Art* (New York: Routledge, 2018), 89.

77 Dekker, *Collecting and Conserving Net Art*, 89–90.

78 Guez et al., 'The Afterlives of Network-based Artworks', 119.

Both Martine Neddam and Annie Abrahams have already opted in for hosting the outdated Flash-based SWF files at the original site of the artwork. In *mouchette.org*, a page called 'Lullaby' consists of a fleeting landscape made up of user contributions, which pass swiftly over a background of dead flies. The animation had originally been scripted by Marc Boon using

Flash-based technology but has, since then, been re-programmed by David Jonas. Nonetheless, the unviewable Flash version of 'Lullaby' is still hosted as a page on the website and is linked in the main animation, despite serving no obvious purpose.[79] Similarly, Annie Abrahams' *Karaoke* still displays the Flash-based animation in all of its greyed-out glory, rather than migrating the page completely to its web-archived version.

Given the discreet but accessible nature of the source code, it becomes a suitable site for implementing obsolete Flash-based components, which artists already seem to be doing, albeit in slight variations. In technical terms, this can be achieved by manipulating the document object model (DOM) of the website's HTML file, which specifies its structure. The artist or conservator can hide the SWF files by setting the display property for the relevant element to 'none'.[80] This will hide the SWF files from the website completely but will nonetheless keep them hosted on the page and make them retrievable through the client-side script. This engenders a non-linear, and to an extent collective, conservation workflow, where users are provided with the opportunity of going back to previous iterations of an artwork, while also re-enacting outdated components in future migrations.

79 See http://mouchette.org/fly/flies.html (accessed 31 May 2021).

80 That is, by inserting the following line of code: 'document.getElementById("divName").style.display = "none";' into the script tag of the final HTML document, where "divName" specifies the name of the division where the SWF files are contained.

Conclusion

This article has outlined a migration method that demonstrates how to apply reverse engineering techniques to reuse core elements in order to ease the migration workflow in Adobe Animate. At the same time, these acts of decompilation invite us to negotiate the tension between the dematerialising and re-materialising narratives in time-based media conservation. As a result of these navigations, a networked object's source code can be appraised beyond its role as a significant original or a score. This opens up new interpretations of the source code where it exists in plurality with its present and past versions, forming a site for open conservation and interaction economies. In a larger sense, this becomes an invitation to reflect on the ways in which certain strategies, including emulation and virtualisation, unintentionally disavow the expanded meaning of community in conservation; and how the source code can recuperate this lost opportunity.

Naturally, there are several limitations that one needs to consider in relation to the proposed method. Because of the laboriousness associated with migration, this type of workflow may be more suitable for individual, targeted projects where there is a demonstrable desire to keep the artwork on the 'live web' or where there is a lack of access to emulation- and virtualisation-based strategies. Moreover, particular artists may be partial towards these strategies because their artwork becomes contained within an isolated container, undermining the conceptual affordances and the viewer's encounter with native elements of the web browser. Besides the importance of applying this method in a suitable context, further research is required to technically evaluate suggestions regarding the use of the source code as a layered site for re-enactment. This may include considering the computational costs of maintaining obsolete files and redundant information at the level of the source code.

ORCID

Anna Mladentseva http://orcid.org/0000-0001-8577-1039

Acknowledgements

I would like to thank the UCL Medical Sciences department for inviting me to take part in a digital preservation project aimed at re-scripting outdated Flash learning resources; this proved instrumental in helping me develop the research presented here. I am also very grateful to have worked on the learning resources project alongside Ricky Wong from the Department of Computer Science, UCL. Finally, many thanks go to the artist

Sinae Kim for our interview, as well as Dr Rebecca Gordon (UCL and University of Glasgow) for supporting me at the early stages of this research.

Résumé

« Répondre à l'obsolescence de l'art numérique construit sur Flash: une étude de cas sur la migration de la *Genèse* de Sinae Kim »

De nombreuses œuvres d'art sur Internet du milieu des années 1990 au début des années 2000 utilisaient la technologie Adobe Flash pour créer du contenu animé. Cependant, à la lumière des récents développements des normes Web (HTML5), Adobe a cessé de prendre en charge Flash et ses outils associés. Le retrait de Flash a rendu des œuvres numériques non fonctionnelles et non visibles, notamment la *Genèse* de Sinae Kim (2001), au centre de cette étude. Les stratégies fondées sur l'imitation—et la virtualisation—récemment proposées ne sont pas toujours adaptées, en particulier si l'on souhaite conserver l'œuvre d'art sur le « web en direct ». Cet article met en évidence la méthode alternative de migration facilitée par les techniques d'ingénierie de conversion —en particulier la décompilation—et met en avant l'importance de maintenir en ligne l'accès aux fichiers Adobe Shockwave Flash (SWF), devenus obsolètes, via le code source. Sur cette prémisse, le code source est réimaginé comme un site de reconstitution ultérieure, permettant de s'écarter de son rôle actuel de marqueur d'« authenticité ».

Zusammenfassung

„Antworten auf die Veralterung Flash-basierter Netzkunst: eine Fallstudie zur Migration von Sinae Kims *Genesis*"

Viele Internet-Kunstwerke der mittleren 1990er bis Anfang der 2000er Jahre nutzten die Adobe Flash-Technologie zur Erstellung animierter Inhalte. Angesichts der jüngsten Entwicklungen des Webstandards (HTML5) hat Adobe jedoch die Unterstützung für Flash und die damit verbundenen Tools eingestellt. Die Entfernung von Flash hat dazu geführt, dass diese Netzkunstwerke nicht mehr funktionieren und nicht mehr angesehen werden können, darunter auch Sinae Kims *Genesis* (2001), das im Mittelpunkt dieser Studie steht. Kürzlich vorgeschlagene Emulations—und Virtualisierungsstrategien sind nicht immer geeignet, insbesondere wenn das Kunstwerk im 'Live-Web' bleiben soll. Dieser Artikel beschreibt eine alternative Migrationsmethode, die durch Reverse-Engineering-Techniken—insbesondere Dekompilierung—erleichtert wird, und stellt die Bedeutung der Aufrechterhaltung des Online-Zugangs zu den veralteten Adobe Shockwave Flash (SWF)-Dateien über den Quellcode in den Vordergrund. Unter dieser Prämisse wird der Quellcode als Ort für eine weitere Re-Inszenierung neu konzipiert, was eine Abkehr von seiner derzeitigen Rolle als Marker der 'Authentizität' ermöglicht.

Resumen

"Respondiendo a la obsolescencia del Net Art basado en Flash: un estudio de caso sobre la migración de 'Génesis' de Sinae Kim"

Muchas obras de arte en el Internet de mediados de 1990 a principios de la década de 2000 utilizaron la tecnología Adobe Flash para crear contenido animado. Sin embargo, a la luz de los recientes desarrollos de estándares web (HTML5), Adobe ha dejado de admitir Flash y sus herramientas relacionadas. La eliminación de Flash hizo que esas obras de arte de la red se volvieran no funcionales y no visibles, incluido 'Génesis' de Sinae Kim (2001), el foco de este estudio. Las estrategias propuestas recientemente basadas en la emulación y virtualización no siempre son adecuadas, especialmente si se desea mantener la obra de arte 'en vivo' por Internet. Este artículo describe un método alternativo de migración facilitado por técnicas de ingeniería inversa, específicamente descompilación, subraya la importancia de mantener el acceso en línea a los archivos obsoletos de Adobe Shockwave Flash (SWF) a través del código fuente. Sobre esta premisa, el código fuente se reinventa como un sitio de representación adicional, apartándose de su función actual como marcador de 'autenticidad'.

摘要

"应对基于Flash的网络艺术过时问题——关于迁移Sinae Kim《创世纪》的案例研究"

自1990年代中期至 2000 年代初期，许多互联网艺术作品使用了Adobe Flash 技术来创建动画内容。但是，鉴于最近的网络标准建设 (HTML5)，Adobe已停止支持 Flash 及其相关工具。Flash的移除使这些网络艺术作品丧失功能且无法查看，包括本次研究重点Sinae Kim 的《创世纪》 (2001)。而新近倡导的基于仿真和虚拟化的策略并不一定适合，特别是如果希望将艺术作品保留在"实时网络"上。本文概述了一种由逆向工程技术（即反编译）促成的替代迁移法，并突出了通过源代码，用以维持对过时的 Adobe Shockwave Flash (SWF) 文件在线访问的重要性。在此前提下，源代码被策划为一个进一步再设定的站点，从而使其脱离当前作为"真实性"标记的角色。

Аннотация

«Поиск альтернатив в связи с устареванием нет-арта на основе Flash: тематическое исследование по миграции произведения искусства Синае Ким „Genesis"».

С середины 1990-ых годов до ранних 2000-ых множество произведений искусства в интернете (нет-арт) использовали Adobe Flash для создания анимаций. Однако, в связи с недавней разработкой открытых веб-стандартов, например HTML5, компания Adobe приняла решение перестать поддерживать программный продукт Flash и соответствующие инструменты. Ликвидация Adobe Flash из интернета привела к нефункциональному и недоступному для просмотра нет-арта, в том числе и произведение искусства Синае Ким «Genesis» (2001), на котором акцентирует внимание данное исследование. Предполагаемые сегодня стратегии, основывающиеся на эмуляции и виртуализации, не всегда являются подходящими, особенно, когда хочется сделать возможным просмотр нет-арта в «естественной среде» интернета. Эта работа излагает альтернативный метод миграции данного произведения искусства путём обратной разработки (декомпиляции) и подчёркивает значимость сохранения онлайн-доступа к устаревшим SWF-файлам (Adobe Shockwave Flash) через исходный код. Из этого следует: исходный код представлен в виде «площадки» для дальнейшей реконструкции («реэнактмент»), и тем самым появляется возможность отойти от его роли как неизменного оригинала.

Biography

Anna Mladentseva is an MSc Digital Humanities student at the University College London (UCL). Having completed her undergraduate degree at UCL's History of Art Department, Anna's research interests operate in the intersections between new media art and digital preservation. She is a member of UCL's Multimedia Anthropology Lab research network, where she has employed three-dimensional modelling, virtual reality and machine learning for ethnographic research. More recently, Anna has been working together with UCL Medical Sciences to re-script the department's Flash-based learning resources.