

**iPRES 2024 • iPRES 2024 Papers**

# **A Condition Assessment and Documentation Framework for Multi- variant Works of Net Art**

**Nicole C. Savoy<sup>1</sup> Dragan Espenshied<sup>2</sup> Martina Haidvogel<sup>3</sup>**

<sup>1</sup>Rhizome/Bern Academy of the Arts, <sup>2</sup>Rhizome, <sup>3</sup>Bern Academy of the Arts

**Published on:** Aug 30, 2024

**URL:** <https://ipres2024.pubpub.org/pub/ruud1i2r>

**License:** [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

*Abstract* – This paper presents a methodology for evaluating the condition of multi-variant net artworks using Rhizome’s Emulation as a Service (EaaS) infrastructure. The proposed three-stage cyclical workflow involves: 1. The collection of evidence of the artwork’s intended performance from existing variants, literature, artist interviews, and data analysis, 2. Building reference variants in EaaS based on the evidence, and 3. The comparison of existing and reference variants’ performances and the documentation of observations in a relational database.

The condition reporting methodology developed from this research revealed a practical documentation framework for managing information regarding complex multi-variant networked digital artworks, using a relational database. This dynamic, open-ended, and cyclical workflow is consistent with the in-process nature of net art and its intrinsic focus on access and performance indicates its potential for an all-encompassing preservation approach.

A version of the database was implemented at Rhizome and is now being applied to their artwork ingestion and documentation process.

*Keywords* – [net art](#), [Emulation as a Service \(EaaS\)](#), [condition assessment](#), [relational databases](#), [knowledge management](#)

*This paper was submitted for the iPRES2024 conference on March 17, 2024 and reviewed by Inge Hofsink, Stefania Di Maria, Leontien Talboom and 1 anonymous reviewer. The paper was accepted with reviewer suggestions on May 6, 2024 by co-chairs Heather Moulaison-Sandy (University of Missouri), Jean-Yves Le Meur (CERN) and Julie M. Birkholz (Ghent University & KBR) on behalf of the iPRES2024 Program Committee.*

## **Introduction**

Due to the diversity and complexity of obsolescent-prone web technologies coupled with relatively recent institutional engagement with net artworks, standards for their conservation have only in recent years begun to be established. The tools and methods used to analyze net art are usually chosen on a case-by-case basis.

The research presented in this paper— conducted within the framework of a master’s thesis at the Bern Academy of the Arts and in collaboration with Rhizome—aimed to develop a broadly applicable methodology for condition reporting multi-variant works of net art. Rhizome is a born-digital non-profit organization founded in 1996 that commissions, presents, and preserves digital art. Its collection of digital art, published in the Rhizome ArtBase, contains more than 2300 works and is growing steadily via artistic programs, community open calls, and institutional partnerships.

To assess the condition of a work of net art in which none or very little of its significant features remain accessible in its current environment (assuming the work's artifacts are available), this paper proposes to build reference variants based on technical and art historical research using Emulation as a Service (EaaS). Subsequently, observing and comparing the performances of existing variants and such reference variants provides a methodology to narrow down the expected performance of the work and the technical environment necessary to yield that performance. Documenting observations and comparisons of multiple variants in an organized and practical format is manageable within a relational database.

## **Networked Digital Objects and Variants**

Variants refer to all existing combinations of one version of a digital object's artifacts (static files, Linux containers, layered disk images, web archives), resources (external links, libraries, etc.), and access / software environments (contemporary or emulated legacy systems) [1][2]. Artifacts can be hosted on a web server and accessed online, within an emulated environment, or as a web archive (Fig. 1). Deliverable artifacts like static files can be made accessible within multiple access environments resulting in different variants of a work. For artworks without deliverable artifacts that retrieve data from remote resources like Google maps, or those that are accessed on platforms like Instagram, artifacts can be constructed by creating web archives of the work's performance. Web archives can be accessed on the public web or in an emulated environment in EaaS [3].

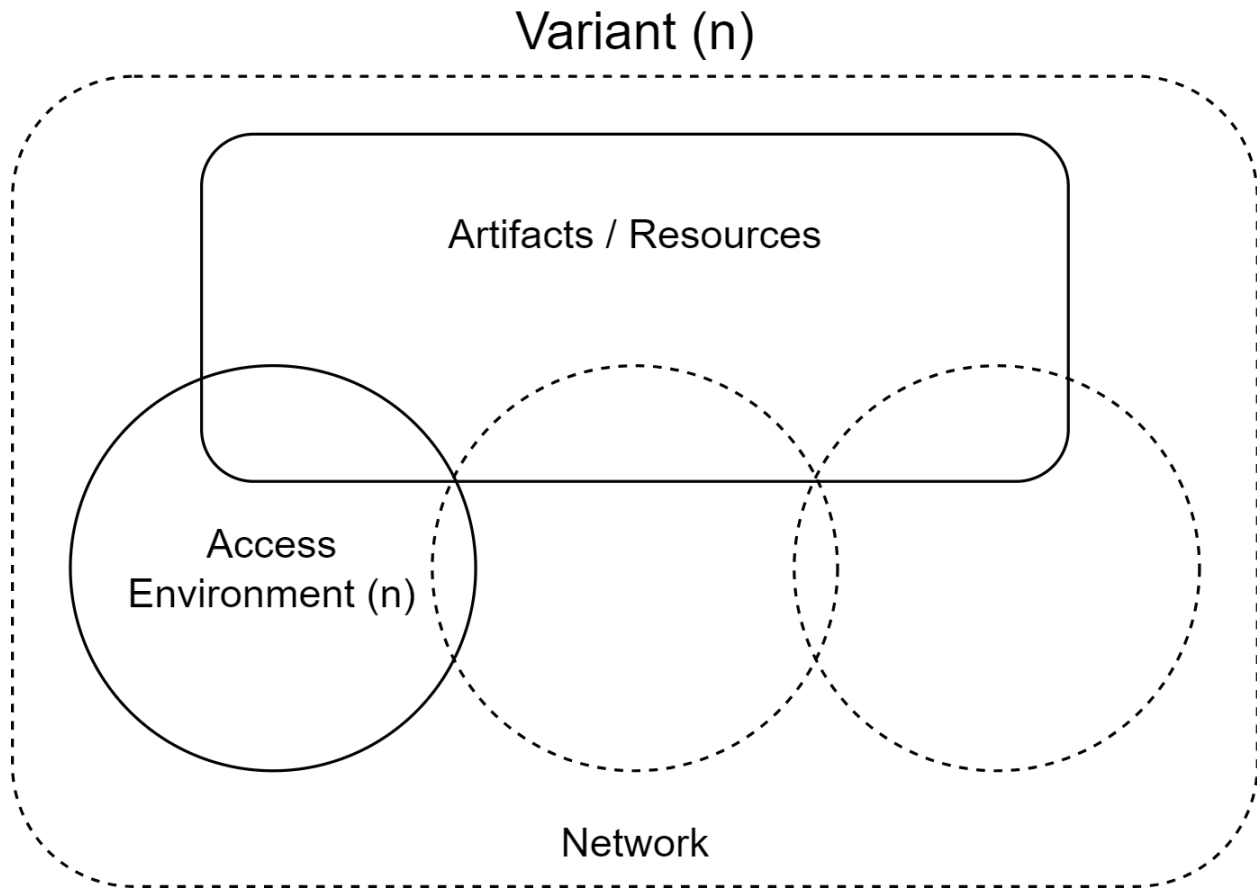


Figure 1: Variants represented as a net artwork's artifacts and resources in an access environment (n) and within a network

Since many net artworks exist as multiple instantiations, or variants, one does not necessarily represent the whole. Rather, the performance of a particular variant is representative of an artwork in a particular technical environment. For example, the artwork *Bodies© INCorporated* by Victoria Vesna (1996-99) currently exists as two accessible variants, on the University of California Los Angeles server (“UCLA variant”) (<http://www.bodiesinc.ucla.edu/>) and also as a partially restored, emulated variant at Rhizome’s Net Art Anthology (“NAA variant”) (<https://sites.rhizome.org/anthology/bodiesinc.html>) (Fig. 2, Fig. 3). Both existing variants are based on the same version of the artwork’s data created by the artist. In the NAA variant, the data is hosted at a Rhizome server and changes were made to restore missing features.

When comparing the two variants, it is apparent that the NAA variant performs many features that are no longer accessible in the UCLA variant. In the comparison of the same page of the site in Fig. 2 and Fig. 3, the work’s 3D features, now inaccessible at its current URL, are accessible in the NAA variant.

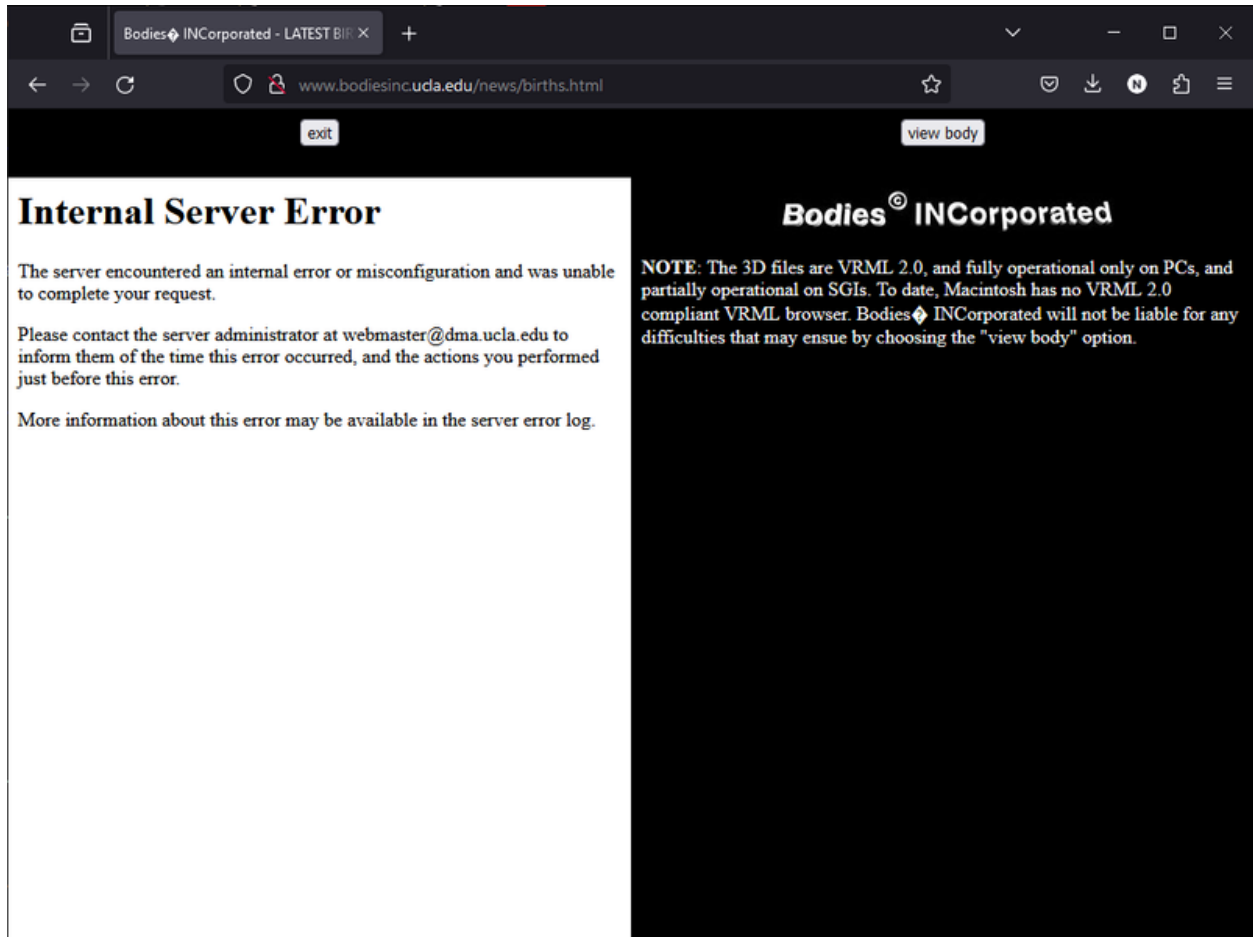


Figure 2: "News" (new body view) server error page. *Bodies<sup>®</sup> INCorporated* (1996-99), Victoria Vesna. UCLA Variant: Windows 10, Firefox 117.0.1 (hosted at UCLA server) <http://bodiesinc.ucla.edu/>

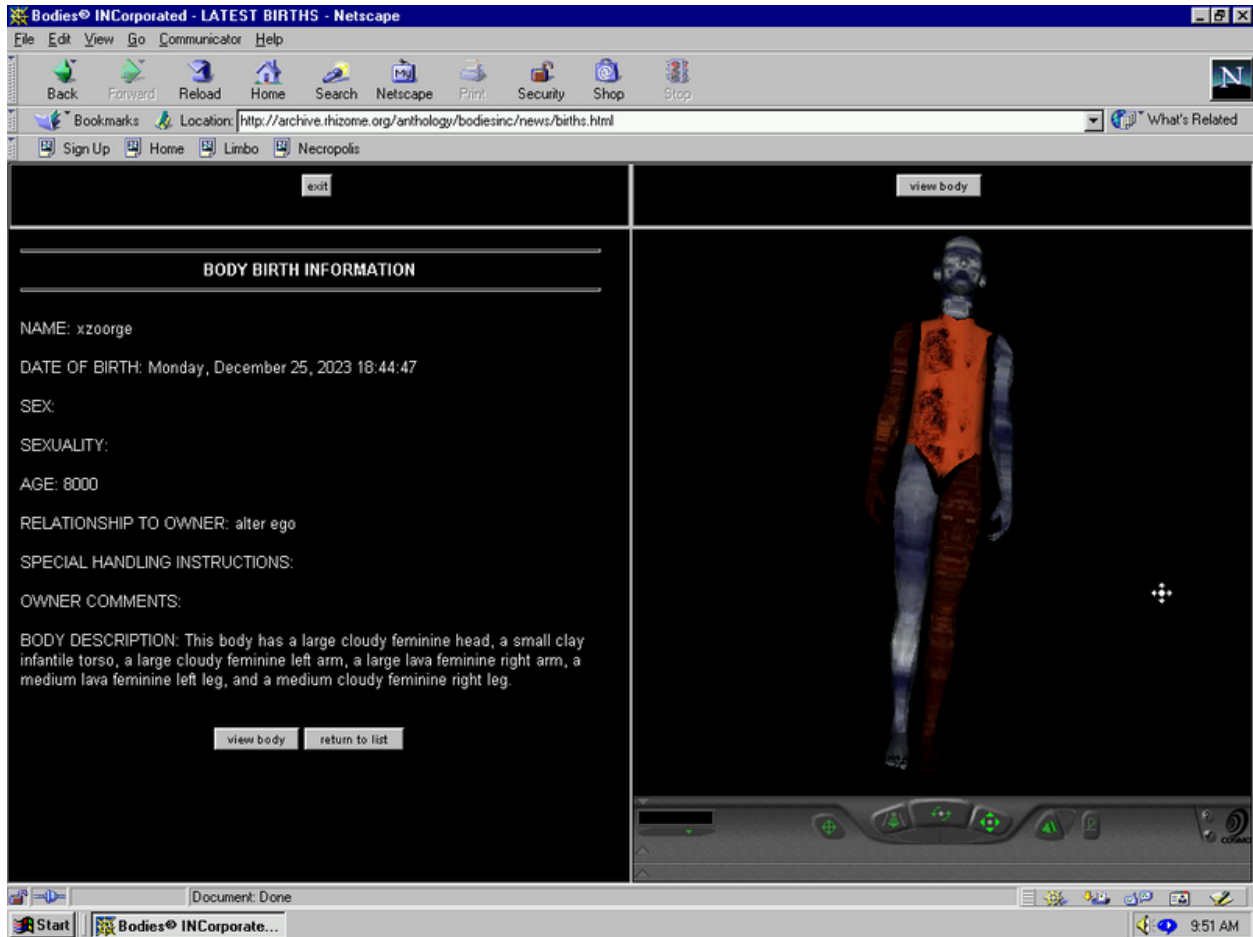


Figure 3: “News” (new body view). *Bodies© INCorporated* (1996-99), Victoria Vesna. Net Art Anthology Variant: Windows 98, Cosmo Player 2.0, Netscape 4.79 (hosted at Rhizome server) <https://sites.rhizome.org/anthology/bodiesinc.html>

As this artwork is exemplary of many of the issues concerning the condition assessment of net art, it will serve as an example going further. *Bodies© INCorporated* exhibits several features common to early net artworks including:

- A vast, complex, and disorganized set of files and directories
- Use of obsolete technology, now inaccessible online
- Little documentation of it performing in its native environment, during its active period
- A multi-year development phase
- No version control system used during its development
- Several ideas and technologies tested that weren't implemented
- Misleading documentation leading to ambiguity of intended performance
- Existing as more than one variant

Victoria Vesna made *Bodies© INCorporated* as a tongue-in-cheek critique of the corporate culture of the 1990s [4]. After agreeing to a set of lengthy terms and conditions, users of the website could become members of a fictitious corporation, build 3D bodies or avatars that they were required to forfeit all rights to, and navigate through the site's 3D worlds.

The work's 3D features were created using Virtual Reality Modeling Language (VRML), a 3D modeling language and matching browser plugins enjoying some popularity around the late 1990s and early 2000s. VRML (sometimes pronounced vermal) was developed in the mid 1990s and was the first internet standard that allowed the creation of interactive, hyperlinked 3D graphics and multimedia content on the web [5].

*Bodies© INCorporated* was one of the few net artworks to use VRML to create interactive 3D content online. The artist and her collaborators developed the complex work over several years while experimenting with the then new technology [6]. Since its active period, most of *Bodies© INCorporated*'s significant features, mainly its participatory and 3D capabilities, have become inaccessible due to VRML's obsolescence and its incompatibility with contemporary browsers. At its current UCLA server, only the work's HTML content is accessible (some of its VRML features were partially restored in EaaS for Rhizome's NAA variant).

The inaccessibility of the artwork's VRML features due to its reliance on specific legacy software and server configurations adds to the already challenging task of discovering how it should function, how it should appear on screen, and how the user should experience it. There is limited documentation of the work being performed during its active period and much of the existing records are misleading. Descriptions of the artwork's VRML features' expected performance are found in literature, documentation and text written on the website itself. However, without documentation of those features performing in their native environment in the form of video recordings, narration, screenshots, or a web archive for comparison, it is difficult to determine how accurate these descriptions are; which of the features are performing as they were meant to, and which are not performing at all. Luckily, for *Bodies© INCorporated*, the existing NAA variant provided a starting point for comparison.

## EaaS

The open-source framework Emulation as a Service (EaaS) was initiated by the bwFLA project (Baden-Wuerttemberg Functional Long-term Archiving and Access) in 2012 with the aim of offering pre-configured emulation environments as a preservation strategy for complex digital objects via a web browser. The EaaS model operates on the premise that as memory institutions collect a more extensive array of vulnerable digital objects, providing user-friendly emulation environments can constitute an effective preservation solution. This approach enables the rendering of these objects in their native environments, facilitating preservation and access to their significant properties. To achieve accessibility to multiple emulation environments on demand, a scalable distributed system model implemented in a cloud infrastructure was developed [7][8].

The fundamental components of the EaaS system consist of emulators and operating systems, functioning as foundational "machines." These serve as the basis upon which emulation environments can be constructed by incorporating software applications as well as the digital object's files and other essential resources or artifacts (Fig. 4) [8].

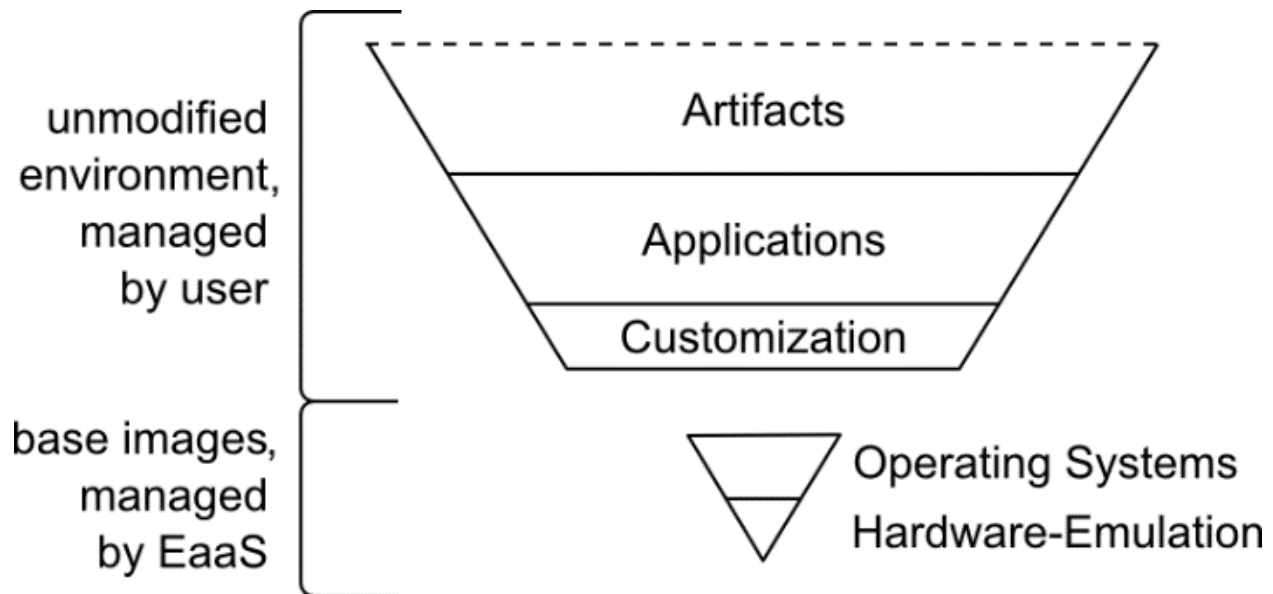


Figure 4: bwFLA Emulation as a Service architecture diagram [8]

Rhizome uses EaaS to preserve and present its vast collection of born-digital artworks. This involves acquiring multiple legacy operating systems, versions of software programs, plug-ins, and other components that can be combined in emulation environments to suit the accessibility and preservation needs of many varied artworks. This approach addresses the technical diversity inherent in net artworks that require highly individualized sets of environments compared to traditional objects. EaaS provides a platform for net art and born-digital art in general to remain accessible by allowing the creation of as many access environments as required.

The EaaS interface makes the process of importing and combining objects (software and other resources) easily manageable. Once objects are imported, they are saved in the system and can be applied to new environments. Instead of building multiple environments from scratch that require the same operating system, each one can be branched from a base with the desired operating system already installed. These new "derivatives" of the base environment can then be customized further to suit different artworks [7].

Each time new software is installed, an object is added, or any other change is made to an environment, it is saved within EaaS as a revision with an option to add a description. New derivatives can be made from any point in an environment's revision history. The ability to create derivatives is especially useful when testing and comparing complex environments built in many steps. To change a component and test the effects on the performance of the artwork being accessed, the environment can be branched at any point in its construction process instead of having to be rebuilt. New objects can be added to the new derivative and differences in

performance can be compared. EaaS' agility to create, modify, and combine possible access environments proved to be useful for building reference variants of *Bodies*© *INCorporated*.

## **Building Reference Variants**

Building reference variants to compare and evaluate artifacts' performances constitutes the foundation of the proposed condition reporting methodology. To arrive at suitable combinations of access environment and server configuration (if applicable) for best performance of the work's features, several avenues of "evidence collection" are pursued. First, data analysis is carried out on available artifacts to determine the work's components, resources, and dependencies (Fig. 5). Additionally, literature, artist interviews, and other forms of documentation are examined to collect further information about the work's performance, albeit limiting and at times contradictory. This in-depth analysis forms the starting point to selecting multiple possible reference variants, combining artifacts with various constellations of environmental parameters. As variants are built, tested, and evaluated, adjustments to their environments are made to narrow down which yield improved performances.

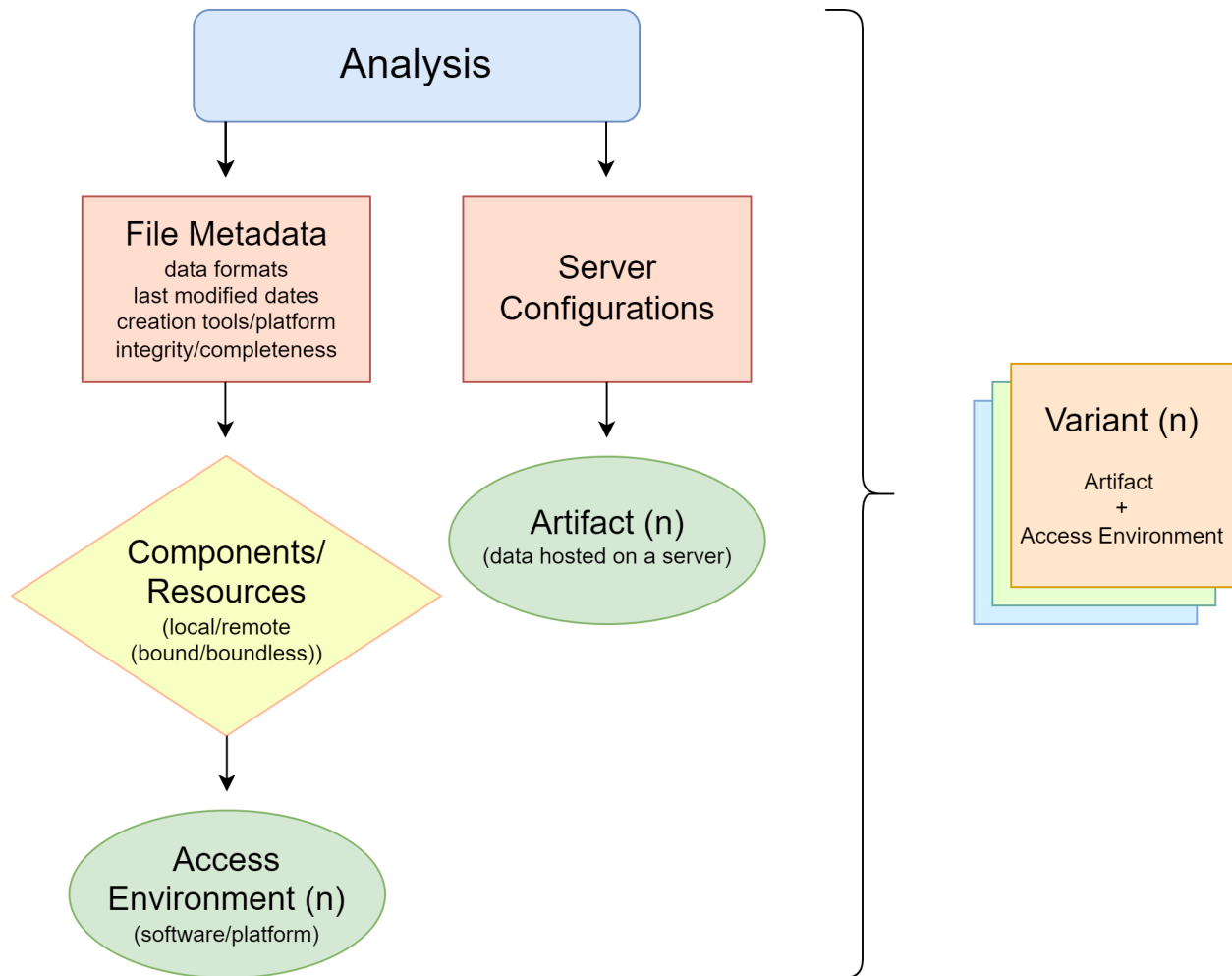


Figure 5: Building reference variant based on technical and art historical analysis

Analyzing the artwork's artifacts or data (if available) with a graphical disk analyzer like WizTree and metadata extraction tools like Siegfried, reveals which file types it is composed of as well as the last modified dates of those files. This information is used to determine which versions of software and server configurations will be tested to perform the artwork's known significant features.

Several versions of *Bodies© INCorporated* existed during its testing and development stage. Since no version control system was used, only one version remains available for analysis. Based on evidence from documentation and the NAA preservation, the presence of VRML (.wrl) files was expected. However, due to the production process that involved testing and experimenting with several ideas that ended up not being implemented, there are files present within the data package handed over by the artist that are remnants of this practice and their performances should not be expected [6].

*Bodies© INCorporated*'s artifacts (mostly static files) are available in their entirety and do not depend on resources that are remote, like hosted software libraries, data retrieved from remote servers, or computational services accessed via API, etc. No material analysis can be conducted on such resources, only their behavior

can be observed and perhaps recorded [3] (An example of a remote resource would be if *Bodies© INCorporated* published videos of generated avatars on a platform like YouTube—the resources that make YouTube work would remain inaccessible, but the platform’s behavior in relation to the artwork could be observed and documented).

*Bodies© INCorporated*’s significant components, the types of files and resources associated with those components, and the software required to access them, is listed in Tab. 1.

Components	Files/Resources	Access Software / Platform
audio	.rpm, .ra	RealPlayer 8
External links	(href tag in code)	Browser with http protocol enabled / Internet Archive Wayback Machine to access archives of broken links
HTML	.html	browser
images	.gif	Netscape 3 or 4 browser
Images	.jpg, .jpeg	browser
Member functions	.txt (database)	.txt file with member info is read and written to by several Perl scripts
Video	.mov	Windows Media Player 6.0
VRML 2.0	.wrl, .gz	Netscape 3 or 4 browser + Cosmo Player 1 or 2 plugin

Table 2: *Bodies© INCorporated*’s variants

The extracted metadata was uploaded to Rhizome’s GitHub account for the artwork where code changes made to the NAA variant were tracked. It was determined that due to updates to the UCLA server over many years, the inaccessibility of several features was caused by changes not only to the access environment but also to the server’s configurations.

The NAA server was coincidentally configured similarly to the artwork’s original server configurations, allowing several features like character encodings and member sign-in functions to work. This server also hosts several other artworks in the Rhizome ArtBase and, therefore, its configuration could not be altered without possibly affecting them. To test the effects of the server on performance, a containerized Apache server was

used that was configured closely to the work’s original server settings. Changes made to the containerized server’s configurations while building reference variants were tracked in GitHub.

The containerized server was used for the Cosmo 2 Container variant and the Cosmo 1 Container variant (Tab. 2). The artwork’s files were put into the containerized server and the environment’s host configurations were set to point to the containerized server instead of the UCLA server. This means that each time any URL associated with *Bodies© INCorporated* is entered into the browser search bar, the artwork is accessed from the containerized server instead of the UCLA server. The UCLA and Cosmo 2 UCLA variants access the artwork’s data from its UCLA server.

As reference variants were built in EaaS, the platform automatically recorded their version histories. No alterations were made to the artwork’s code in the reference variants; changes were instead made to their access environments and server configurations (containerized server). The resulting variants to be compared included the two existing variants, UCLA and NAA, and three reference variants built in EaaS (Tab. 2).

Variant	Access Environment	Artifacts (files hosted on a server)
UCLA	Windows 10 + Firefox 102-121	<i>Bodies© INCorporated</i> at UCLA.edu
NAA	Windows 98 + Cosmo Player 2 + Netscape 4	<i>Bodies© INCorporated</i> at Rhizome server
Cosmo 2 Container	Windows 98 + Cosmo Player 2 + Netscape 4	<i>Bodies© INCorporated</i> at containerized server
Cosmo 2 UCLA	Windows 98 + Cosmo Player 2 + Netscape 4	<i>Bodies© INCorporated</i> at UCLA.edu
Cosmo 1 Container	Windows 98 + Cosmo Player 1 + Netscape 4	<i>Bodies© INCorporated</i> at containerized server

## Condition Assessment

The condition assessment of net art is focused on the performance of data. However, the condition of data (artifacts) as objects can also be assessed. Unavailable or corrupt data is considered loss. Assuming the data is available and complete, condition reporting revolves around several variants in comparison, rather than one single object. As variants are compared, observations of their performances are recorded (Fig. 6). This cyclical approach of comparing multiple variants, modifying reference variants to access inaccessible features, and documenting observed performances, narrows down which performances should be expected.

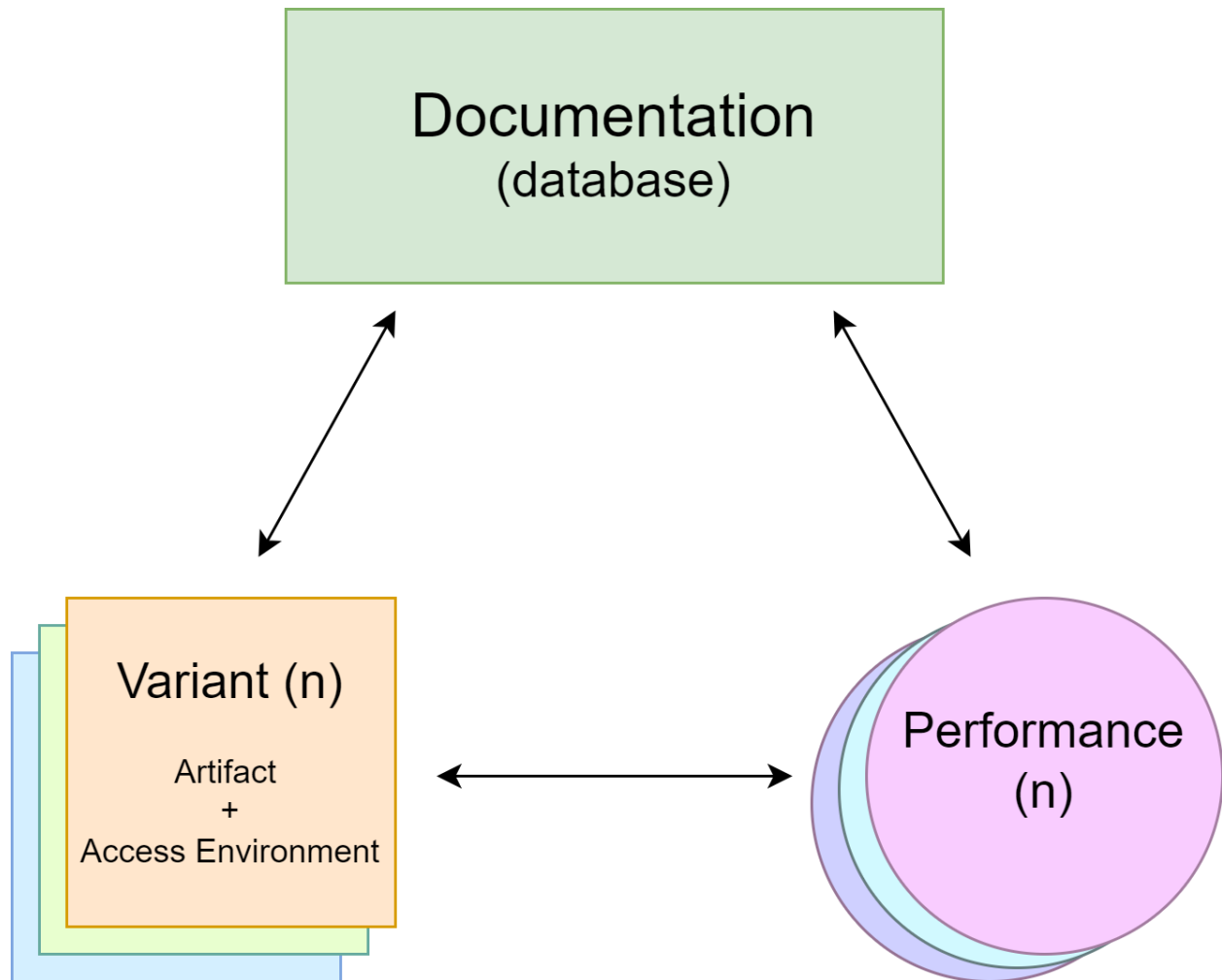


Figure 6: Condition reporting workflow

The workflow focuses on observations made while comparing variants' performances. The neutral term "observation" is used instead of error or issue because it is often unclear initially until a cause is determined if an observed performance is intended or not. "Evidence" regarding the work's significant features is collected and linked to related observations, as well as "questions" relating to observations, "causes" of observed performances, and potential "preservation actions". Evidence gathered from various sources including documentation, literature, artist interviews, and data analysis can clarify the work's components, significant features, and expected performances. Evidence also leads to known causes for observed performances. Depending on the observation, a known cause can confirm whether an observed performance is intended or not (error or non-error). Although this paper concentrates on condition assessment, the inclusion and interdependence of analysis, documentation, and possible preservation actions derived from performance tweaks, highlight this workflow's potential as an all-encompassing preservation approach.

## Initial Comparison: Existing Variants

An initial comparison of the two existing (UCLA and NAA) variants showed that the NAA preservation was successful in reestablishing several features that had stopped functioning on the UCLA site (Fig. 2, Fig. 3). Some comparisons led to clear answers of an observation's cause and were as simple as accessing the work in the appropriate legacy browser. By accessing the work at the UCLA server in EaaS with a Windows 98, Cosmo 2.0, and Netscape 4 access environment, some features that are now inaccessible online, were observable. This comparison revealed that the access environment alone was responsible for those aspects' performance and that the others required further investigation. To examine more complex problems, additional references would be needed that can perform lost features. By comparing a variant where a certain feature is not performing with a variant in which the feature does perform, causes and potential preservation actions can be determined.

## EaaS Reference Variant Comparison

Based on the NAA preservation, it was already known that Cosmo Player 2.0 performs *Bodies*© *INCorporated*'s VRML features (Fig. 7). Several hints in the work's documentation and on the site itself suggest that Cosmo Player 1.0 was implemented at some point. For this reason, a reference variant with this version of the software was built for comparison. However, it was discovered in an interview with the artist and one of her collaborators that Cosmo Player 1.0 was indeed tested in the early development stages of the artwork, however the website was built to work with Cosmo Player 2.0 [6]. In the Cosmo Player 1.0 reference variant (Fig. 8), some VRML features are rendered but the site crashes soon after entering the 3D worlds. This variant was, therefore, no longer used for comparison.

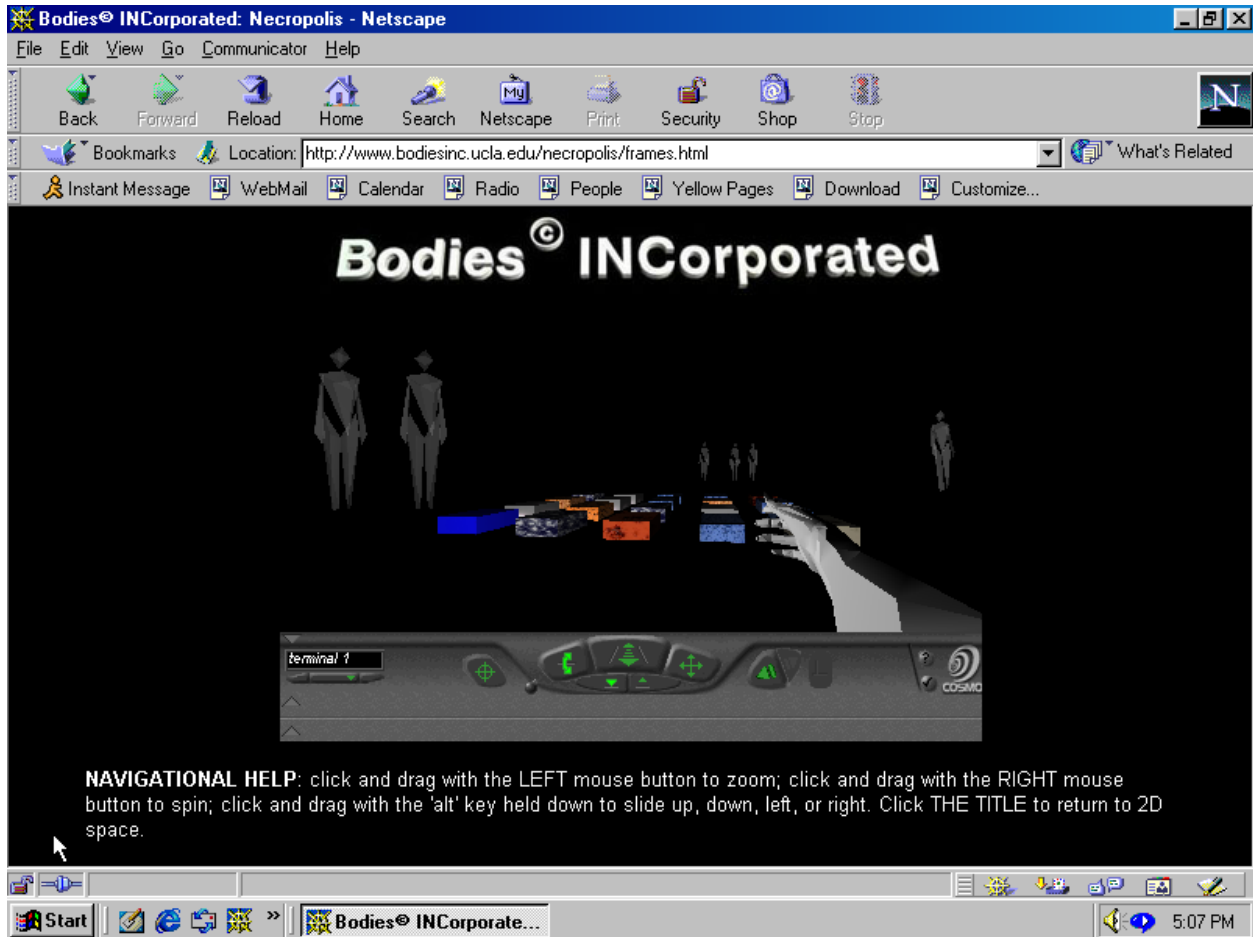


Figure 7: “Necropolis” 3D world. *Bodies© INCorporated* (1996-99), Victoria Vesna. EaaS Variant: Windows 98, Cosmo Player 2.0, Netscape 4.79 (hosted at a containerized Apache server)

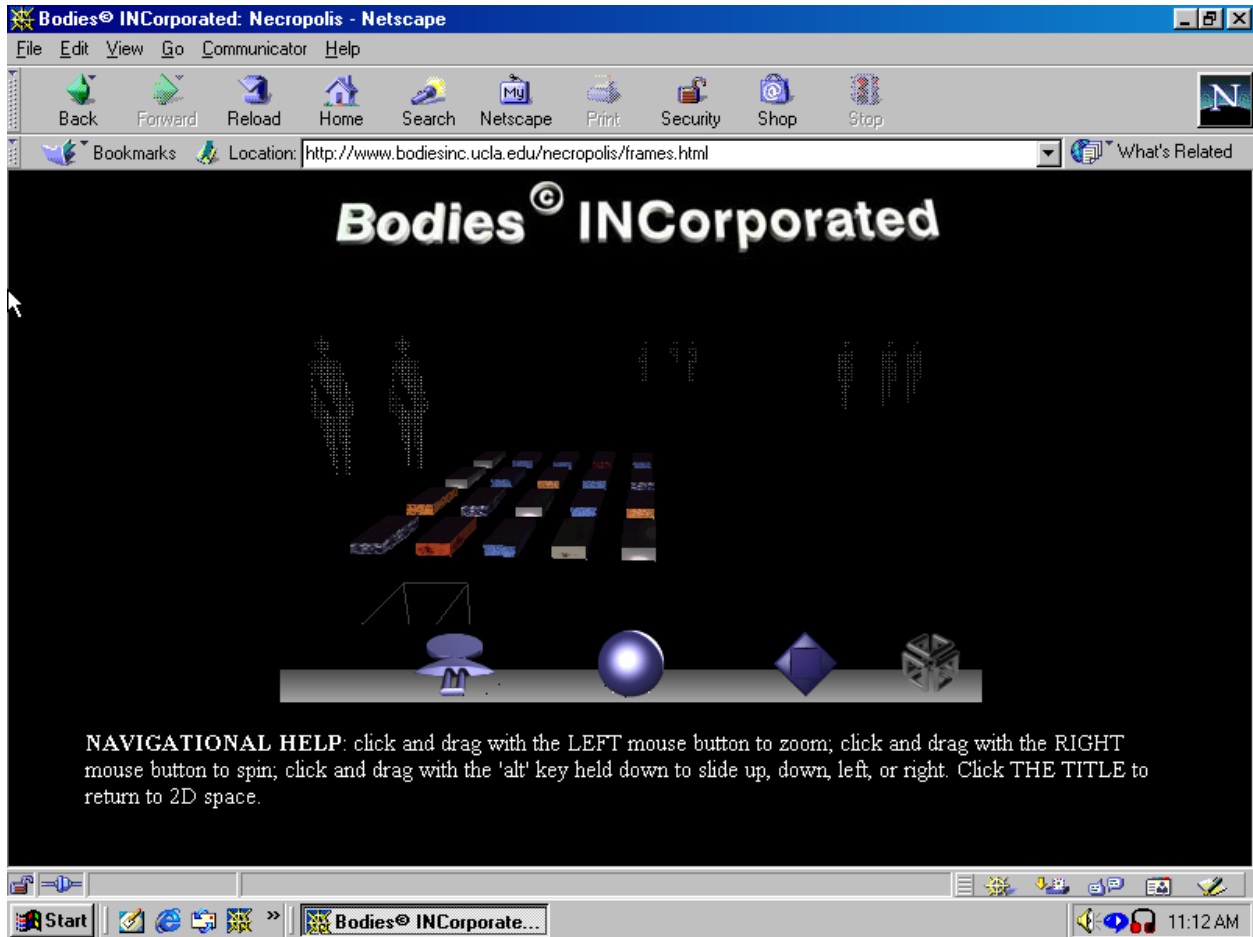


Figure 8: “Necropolis” 3D world. *Bodies© INCorporated* (1996-99), Victoria Vesna. EaaS Variant: Windows 98, Cosmo Player 1.0, Netscape 4.79 (hosted at a containerized Apache server)

## Documentation

The variant comparison examples described in the last section illustrate the potential complexity of documenting such a process. Dozens of observations were made while assessing *Bodies© INCorporated*. To systematically approach the large amount of information being collected while comparing variants, document observations of their performances, and to refine how the data was recorded, organized, and accessed, a relational database was built (Fig. 9). Within the structure of the database, observations were recorded as text descriptions, screen shots, and screen recordings. Links to other related individual records, such as components of the work; variants in which an observation is occurring, not occurring, or that have not been tested; questions regarding an observed performance; and written descriptions of evidence, causes, and preservation actions were also included in the observation records. Different views were made for specific workflow processes. For example, when comparing observations amongst variants, only the observations, variants occurrence, variants no occurrence, and variants not tested properties are shown. Other views group

observations by priority, confirmed error or non-error, known or unknown cause, re-examination required, solution found, and those dealing with specific variants or components.

Employing a relational database in such a way, facilitates workflows that are used to treat works and manage the information gathered during those processes. Its dynamic open-ended structure is integral to the condition assessment workflow and especially practical for net artworks that are in-process by nature and existing as multiple variants.

The database that was developed for this research was built with the workspace platform Notion which Rhizome uses for their project management. Several open-source options exist that make building similar database templates possible. One such platform, Airtable, was used successfully to recreate the database (Fig. 9). Since it can be shared and customized, the Airtable template is a practical solution for collaboration on projects utilizing the database.<sup>1</sup>

Observation	Description	Media	Status	Variant Occurrence	Variant No Occurrence
1 VRML features are not accessible	None of the artworks 3D VRML features are accessible in variants where the artwork's data is hosted at the UCLA server. VRML access is dependent on the combination of access ...		Confirmed Error	EaaS_UCLA_Reference_Variant_03 UCLA_Existing_Variant_01	EaaS_Container_Reference_Variant, NAA_Rhizome_Existing_Variant_02 EaaS_Container_Reference_Variant,
2 Cosmo 1 crashes	cosmo 1 crashes when accessing 3D worlds		Unknown Cause Confirmed Error	EaaS_Container_Reference_Variant,	EaaS_Container_Reference_Variant,
3 Arm (not whole body) visible in Cosmo Player	In the video footage from 1997 of the artwork performing on an SGI computer, a whole body is visible in the 3D spaces. However, in the variants where VRML is accessible via Cosmo ...		Confirmed non-Error	EaaS_Container_Reference_Variant, EaaS_Container_Reference_Variant, NAA_Rhizome_Existing_Variant_02	

Figure 9: Screenshot of a section of the workflow database in Airtable

The transferability of data across platforms is an important issue to consider as their longevity is not guaranteed. Other tools like Obsidian, which is based on a simple, durable plain text file format, were also evaluated. Tests were made to transfer the database from Notion into Obsidian and Airtable. Although several import and export options exist for each platform, the imported database required several adjustments in Obsidian and Airtable. These options may improve as the platforms themselves develop. However, for a large database with several interrelated records, transferring across platforms does not currently seem to be a practical solution.

## Rhizome Implementation

The increasing number of born-digital artworks that require preservation treatment not only necessitates to build technical infrastructure, such as Emulation as a Service that can perform legacy software at scale, but also calls for developing better ways to continuously create and manage knowledge about artworks, in particular, reasoning and research that led to the execution or dismissal of certain preservation actions. Already in early 2021, Rhizome had moved away from working with documents, spreadsheets, and reports to using relational databases for artwork preservation and documentation, with a classic kanban model for work planning and task management. While this standard approach to managing tasks introduced many practical benefits to preservation work, it soon became apparent that critical information about the reasoning that led to the creation of each task couldn't be adequately connected and represented in a way that would keep that knowledge easily retrievable and actionable over time.

The structuring of condition assessment into discrete activities and resources—observations, evidence, causes, artwork components, questions, and preservation actions—as introduced above, allows preservation staff to capture that information. Additionally, knowledge that is applicable across several technically or periodically similar artworks can be related across artworks and serve as a blueprint for analysis and treatment. For instance, techniques for the analysis of works created using a particular programming language are typically useful to apply to multiple works; an access environment created for a particular artwork might be used as a testbed for a contemporaneous work or become a starting point for a new variant.

To achieve this level of information organization, the design of this project's prototype database was expanded with a new table that lists all artworks, and a new reference property was added to existing tables pointing to the artwork that the observation, evidence, causes, etc. applies to. As such, the data for one or more artworks can be accessed by creating different views into the complete dataset, which, over time, will accumulate knowledge about a growing number of works in the collection. This is also helpful when Rhizome performs artwork preservation services for outside clients: the knowledge gathered on works in the collection can be applied to and merged with knowledge gathered on client-owned works.

## Conclusion

The condition assessment methodology presented in this paper demonstrates a practical approach for assessing complex net artworks and is effective in identifying inaccessible features for works where no conclusive references exist. The process of collecting evidence and analyzing the work's data, using that evidence and analysis to develop references, and comparing reference variants and existing variants, successfully clarifies intended and unintended performances and proved to be well suited for multi-variant networked digital objects.

The duration of the overall assessment is dependent on the complexity of the work, the depth of the research, and the potential observations that might be raised to encountered issues. These variations make determining when the process is or will be finished only discoverable within the workflow itself. The condition assessment

of *Bodies*© INCorporated lasted several months. The scope of the master's thesis allowed for thorough and in-depth research, and the case study's vastness and complexity created numerous avenues for exploration into its technical composition. Its many components and features provided multiple diverse observations whose causes are equally varied. Observing and comparing variants' performances revealed causes, some that led to preservation solutions and others that required further investigation.

The use of EaaS as a condition assessment tool was an essential part of the workflow, as it provided a platform for creating, testing, and comparing multiple reference variants. Instead of building them from scratch each time, the use of base environments and the ability to build new branches from an emulation environment at any point in its revision history simplified the construction of new reference variants. This enabled quick testing and comparing of the performative effects of specific environmental changes.

The database documentation tool functions as an optimal platform for the collection and organization of information, whether it be from analysis, condition reporting, research, or any other sources. The dynamic structure of the relational database simplifies documentation of observations and other record entries linked to specific variants and components of the work. Each entry's status is tracked, and those relating to specific observations, components, variants, etc. can be quickly located for examination. The database structure facilitated a clear outline of observations made under specific conditions, guiding determinations of causes for variants' observed performances. The information collected is navigable within this system, offering valuable guidance for further research. Within the dynamic, open-ended database, the condition reporting process is easily organized and maintained. This platform supports the distribution of the workflow among stakeholders and enables timestamping of all activities in the database assigned to those involved within each entry.

### ***Limitations to the Database***

Notion was used to build the database because it is used by Rhizome. The longevity of platforms like Notion is not guaranteed and backup solutions need to be established. Other similar free and open-source tools exist, such as Obsidian and Airtable, and transferring data across platforms is possible, although not seamlessly. Users interested in keeping their data private can watch the development of open-source projects like AppFlowy, Baserow, nocoDB, or the free-to-run SeaTable, all of which provide similar functionality to Notion and AirTable, but can be deployed in-house instead of being used as a cloud service. As these projects continue to develop, further integration capabilities like tracking and linking to variant revisions in EaaS and alterations, like server configurations, tracked in GitHub would improve interlinking documentation workflows even more.

A database like Notion is not integrated with widely used museum collection database systems like TMS or MuseumPlus and needs to be managed independently. Adding an extra workflow tool that, depending on the collection, may only serve a small subset of artworks may be met with some resistance. And there is certainly an argument to be made to not create knowledge silos in the process. The complexity of multi-variant artworks, however, requires specialized expertise, approaches, workflows, and tools to effectively meet their preservation

needs. Complementing and expanding existing documentation systems may, in turn, benefit institutional knowledge management and artwork stewardship overall.

## Footnotes

1.

The Notion database for *Bodies*© *INCorporated* can be viewed at the following link:

<https://rhizomedotorg.notion.site/Condition-Reporting-Research-d4b868e8b729496c97b92d9965921eaa>

The Notion database is noneditable and is for research purposes only.

An editable Airtable template of the database can be copied at the following link:

<https://airtable.com/appBgXUvTolFVZDOP/shr3VfkF7gM0Kzpsr/tbl1qO80sJOet9cQg/viwLurs7mxrrHCEkU?blocks=hide>

The authors encourage experimentation with the workflow using the Airtable database. ↵

## References

1. D. Espenschied, “Artifacts and Infrastructure,” *Rhizome Almanac*.  
<https://almanac.rhizome.org/pages/artifacts-infrastructure> (accessed on 12/23/2023). ↵
2. X. Ma, D. Espenschied, and L. J. Moulds, “Access Quality Metrics for Net Art,” *iPres 2022*, 2022, pp. 250–254. DOI 10.17605/OSF.IO/6RNK4. ↵
3. Espenschied, D, “Digital Objecthood,” in Selçuk Artut, Osman Serhat Karaman, Cemal Yılmaz, eds. *Technological Arts Preservation, Sabancı University, Istanbul, 2021*, pp. 117-139, ISBN 978-625-7329-16-3. ↵
4. V. Vesna, *Database aesthetics : art in the age of information overflow*. Minneapolis: University Of Minnesota Press, 2007. ↵
5. R. Carey and G. Bell, *The Annotated VRML 2.0 Reference Manual*. Addison-Wesley Professional, 1997. ↵
6. V. Vesna and N. Freitas. Personal communication [Interview]. 2023. ↵
7. E. Cochrane, K. Rechert, S. Anderson, C. Fox, J. Oberhauser, and E. Gates, “Useable Software Forever: The Emulation as a Service Infrastructure (EaaS) Program of Work,” *iPres 2022*, 2022, pp. 40–52. DOI 10.17605/OSF.IO/AY48C. ↵
8. “bwFLA: Emulation as a Service,” [web.archive.org](http://web.archive.org), May 18, 2020.  
<https://web.archive.org/web/20200518112558/http://eaa.uni-freiburg.de/> (accessed Dec. 16, 2023). ↵