

# A Case Study on Emulation-based Preservation in the Museum: Flusser Hypertext

Frank Padberg

Karlsruhe University of Arts and Design (HfG)  
76135 Karlsruhe, Germany  
fpadberg@hfg-karlsruhe.de

Daniel Irrgang

Berlin University of the Arts (UdK)  
10823 Berlin, Germany  
irrgang@medienhaus.udk-berlin.de

Philipp Tögel

Berlin University of the Arts (UdK)  
10823 Berlin, Germany  
variantology@digital.udk-berlin.de

Martin Häberle

ZKM Center for Art and Media Karlsruhe  
76135 Karlsruhe, Germany  
martin.haeberle@zkm.de

## ABSTRACT

We use emulation to preserve a complex digital artifact in the museum. We describe all stages of the preservation process and discuss the technical and curatorial problems that we encountered. The exhibition setting defines additional requirements for an emulation. Our findings and lessons learned are instructive for emulation researchers and museum practitioners. The preserved artifact now is on display in real exhibitions.

## Keywords

Emulation; Digital Preservation; Digital Art; Museum Exhibition; Virtual Disk; Hypertext Systems; HyperCard; Vilém Flusser; Media Theory

## 1. INTRODUCTION

We report on a successful effort to preserve a complex digital artifact of the early 90s and present it in a museum exhibition using emulation. We describe the instructive difficulties that we encountered at all stages of the preservation process, ranging from the preparation of the digital artifact to the operation of the emulation in the exhibition. We also comment on the technical and curatorial decisions that we made.

We suppose that our observations and findings are quite typical for an emulation-based approach, and that the lessons learned in our case study will prove useful for emulation researchers and museum practitioners.

The digital artifact that we aimed to preserve is the so-called “Flusser Hypertext” from 1992. This work not only is part of the digital heritage of the philosopher and media theorist Vilém Flusser, but also an important document of the early history of hypertext systems. The Flusser Hypertext project started at a time when the Web as we know it today was still in a very early phase of its development. Few guidelines had been established at that time for how to structure and lay out hypertext content, and a lot of experimentation was going on.

The Flusser Hypertext lent itself to an emulation approach since it originally executed on a standard computer of its time for which an emulator is available “off the shelf,” and it had no special hardware or interface requirements. An earlier attempt to display the Flusser Hypertext at the art festival “Transmediale 2010” using a vintage computer had to be suspended, because the old hardware turned out to malfunction too frequently when operating over days. Hence, this time we opted for an emulation of the vintage environment on modern, reliable hardware.

The preserved Flusser artifact was actually presented in public as part of the retrospective exhibition “Without Firm Ground – Vilém Flusser and the Arts,” shown from August to October

2015 at the ZKM Center for Art and Media in Karlsruhe, and from November 2015 to January 2016 at the Academy of Arts in Berlin. The exhibition is currently on display until May 2016 at “The West,” a gallery and art museum in Den Haag, Netherlands.

The emulation proved stable during its many weeks of operation in the exhibition. Concerning the preservation process, main summary findings are:

- The lion’s share of the effort went into the analysis of the run time environment required by the artifact, and the preparation of the virtual disk.
- The preservation required deep technical knowledge and definitive curatorial judgments at all stages.
- The exhibition setting posed additional challenges for the emulation.
- True to original hardware was a valuable, sometimes indispensable tool in the preservation process.

This paper presents a single-case study, but our findings are supported by our previous and ongoing experiences with preparing multimedia and digital art objects for an emulation-based presentation in the museum.

## 2. RELATED WORK

Emulation has been studied and discussed for 20 years [1] as a preservation technique for digital objects, and it already was the subject of substantial research efforts (see, f.e., [2][3][4]). Yet, emulation seems to have been applied mainly to multimedia in libraries [5][6] and computer games [7][8], but not much in software-based art. There also seems to be a technical research focus on automated emulation frameworks [9][10][11][12][13].

In the art museum, emulation seems to have been employed only rarely as yet. As a result, there is a lack of concrete observations from real exhibitions, and the amount of practical advice available to museum practitioners is very limited.

Rinehart and Ippolito [14] report on a symposium and exhibition in 2004 centered about various uses of emulation in games and art. Among other things, three pieces of software-based art were actually emulated one-one. The emulations were compared against the originals, and the impact of emulation on the appearance of the art works was discussed.

Kaufmann [15] in 2011 describes the hardware emulation of a home computer cartridge that contained a rare, but important software art object; the emulated cartridge was used in the subsequent exhibition. Padberg [16] in 2014 shows a software emulation of the same art object in contrasting juxtaposition with a true-to-original version of the work. Falcao e.a. [17] in 2014 briefly sketch an in-house, exploratory trial to preserve two digital objects from their art collection using virtualization.

The online museum rhizome.org [18] presents a growing number of digital objects using emulation. Their emulations must run over the Web and, hence, are restricted to objects that have minimal interface requirements, requiring just a display, keyboard, mouse, and sound. Clearly, presenting online is much different from the typical museum exhibition in real spaces, and this fact has a strong impact on the look-and-feel of the emulated art object. Similarly, Espenschied e.a. [19] in 2013 present a case study in which six selected software-based artifacts of the “Transmediale” art collection on CD were emulated online using their web service-based emulation framework.

Lurk, e.a. [20] in 2012 discuss requirements for the emulation of digital heritage objects, including digital art. Lorrain [21] very briefly reports on a failed emulation of a software-based art work in 2010. Besser [22] in 2001 discusses the effect of exchanging vintage I/O devices against modern ones on the appearance of digital art objects.

### 3. ARTIFACT

The Flusser Hypertext is based on the 1989 lecture “Schreiben für Publizieren” [Writing for Publishing] by the Czech philosopher and media theorist, Vilém Flusser (1920-1991) ([23] p.510). The lecture was given at the Institute for Technology Assessment and Systems Analysis (ITAS) of the Karlsruhe Nuclear Research Center (today part of the Karlsruhe Institute of Technology).

The Flusser Hypertext was developed as part of the ITAS research project “Elektronisches Buch“ [Electronic Book]. The project goal was to conduct research on the conceptual and technological possibilities of an “innovative electronic presentation form for results of scientific projects”<sup>1</sup> [24]. The ITAS team envisioned to develop a “multimedia study system” that would use Flusser’s lecture (which was accessible as both audio recording and text) as a starting point, and to expand it with additional information and interactive elements.

To transform Flusser’s lecture into a multimodal hypertext means to take Flusser’s theories serious: transferring the spoken word into the electronic text domain, while enriching it with other forms of media (sound, images) and interactive elements [25]. Flusser himself described multimodal electronic media as transitional phenomena, preparing the dawn of an “universe of technical images” [26]. By building upon his earlier language philosophy and communication theory, Flusser’s media theory of the 1970s and 1980s can be read as an analysis of the coming informatized society, claiming (similar to Marshall McLuhan) the end of writing as the dominant discursive form. According to Flusser, written text and the “linear” structure of discourses will vanish and soon be replaced by what he called “synthetic images”: visualizations of concepts that need not be transcoded into letters but can be “calculated” as computer generated images.

The Flusser Hypertext was developed by the team of the ITAS (Knud Böhle, Ulrich Riehm, and Bernd Wingert) and a team of freelance programmers. Vilém Flusser was not directly involved in the development. However, he did supply further information on and explanations of references that he made during his lecture. Bernd Wingert demonstrated an early version of the Hypertext in May 1991 at Flusser’s home in Robion, France (cf. [24] p.209 and [27] p.109). According to Bernd Wingert<sup>2</sup>, Vilém Flusser was obviously honored to see his

words being adopted by the technological apparatuses which he had been theorizing about for years.

The Flusser Hypertext was never finalized or officially published. The version discussed in this paper reflects the last state of the project work from 1992. Live versions of the Hypertext prototype were demonstrated and discussed at six different public venues between 1990 and 1993 ([28] cf. [29]; [30] cf. [24] p.161; [31]; [32]; [33] cf. [24] p.194; [34][35]).

The Hypertext was programmed using Apple’s “HyperCard” system, an early, general-purpose authoring system that supports multimedia content and allows for programmed layouts. The backbone of the work is the transcript of Flusser’s lecture. The transcript is organized as a so-called stack of cards, that is, the text is subdivided into slices that fit onto a single, fixed-size HyperCard screen. The cards are linked by number, and the user can browse through the cards by clicking on the numbers at the bottom and right edge of each card, see Figure 1.

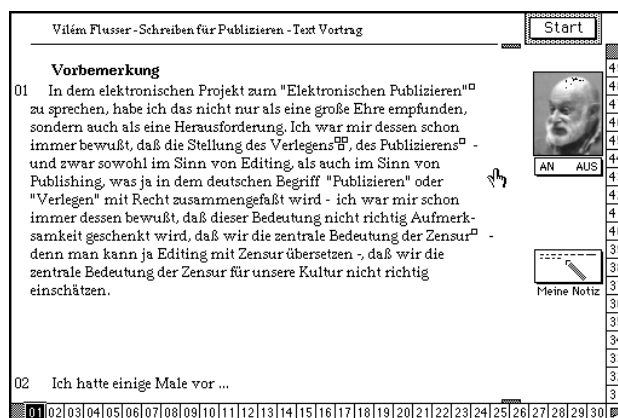


Figure 1. Sample text card

When audio gets activated by the user, the text cards are underlaid with the live recording of Flusser’s lecture: The fragment of the recording corresponding to the current card is played, and at the end of the card the system automatically moves to the next card.

The transcript is augmented by hypertext links that open separate cards containing bibliographical references, short articles, or annotations explaining particular topics and names mentioned in the lecture, similar to today’s web links. The articles were provided by experts in the field, including Flusser himself. This supplementary material (about 450 cards) far exceeds the main text in size (49 cards). Additionally, users have the option to add their own notes to any card: The note pad opens; any text will be saved automatically and can be edited later.

Overall, the Flusser Hypertext is clearly structured. There is no poly-hierarchical network of links but a structure which could be described as “horizontal and vertical” ([24] p.187-188): The cards in horizontal order present the lecture, the cards in vertical order contain the supplements; see Figure 2. Each vertical “string of cards” is separated from other vertical strings – there are no links connecting articles or annotations from different text cards.

<sup>1</sup> all quotes were translated from German by Daniel Irrgang

<sup>2</sup> personal conversation with Daniel Irrgang and Philipp Tögel

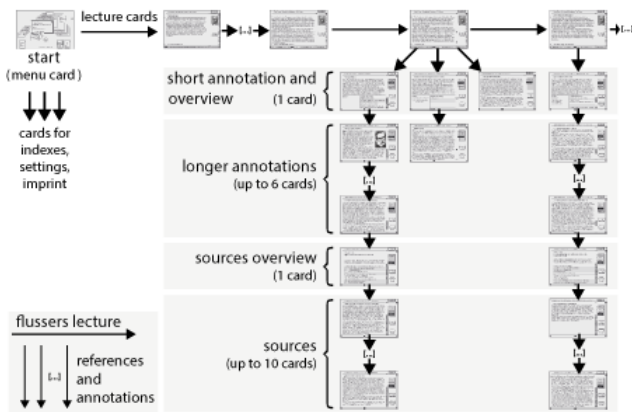


Figure 2. Link structure of the Hypertext

As Peter Wiechens concludes in a case study [36] on the Flusser Hypertext: The Hypertext does not put an end to the lecture’s linear structure but rather adds a dimension of simultaneity due to its multimodal features (text, image, sound). Simultaneity and multi-modality, as well as the “instantaneous availability” of further information (the annotation and reference cards) are pointed out as the Flusser Hypertext’s pivotal aspects in the ITAS research report [24]. One might add to this list the possibility to add own notes to each card. For Vilém Flusser, this kind of productive engagement – the reader being put in the position of an author – was certainly a striking feature that he emphasized in his reflections on the Flusser Hypertext [37]: “The original lecture level falls into oblivion, overwhelmed by comments and counterarguments.”<sup>1</sup>

#### 4. PREPARATION

For an emulation, a so-called virtual hard disk is required as input. This is a file (on the host computer) that has the same internal structure as the hard disk of the vintage, to-be emulated system. The virtual disk also contains a copy of the vintage operating system, which will be booted automatically by the emulator.

There are basically two ways to obtain a virtual disk: either by using a tool that creates a file with such a structure and allows for installing the vintage operating system on this file; or, by taking a one-one file image of an existing vintage hard disk. In our case study, we started with a hard disk image.

The digital artifact can be put in two places: either it is placed in a separate file having a standard format such as an ISO file that serves as additional input for the emulator; or, the artifact’s files get copied to the virtual disk. In our case, the hard disk image already contained the Flusser Hypertext files.

Although we started from a one-one image of a real hard disk, we encountered technical problems when preparing a virtual disk for the emulation. We had to take additional measures to adjust the run time environment of the Flusser Hypertext on the virtual disk. Exceptionally for an emulation, we even patched the code of the artifact. We also had to de-activate certain extensions of the vintage operating system so that the emulator booted the virtual disk. This is a common problem, since many emulators do not support all features of a vintage system.

#### 4.1 Creating the Initial Virtual Disk

The Flusser Hypertext is a research prototype that never advanced into a production-quality release. Only a single copy of its code and data files still existed, stored as a snapshot of the last development version<sup>3</sup> on the physical hard disk of a vintage Macintosh Performa 630 computer. The vintage Mac was donated to the Flusser Archive in Berlin in 2007.

As the first step in the preparation process, which began in June 2013 at the Flusser Archive in Berlin, we created a one-one image of the vintage hard disk and stored it in a file. Detaching the hard disk from the vintage computer and attaching it to a standard PC usually is the best option, see Figure 3.



Figure 3. Cloning the hard disk<sup>4</sup>

The image file contains both the vintage operating system and the Flusser files and, basically, can serve as the virtual disk for the emulation. When trying to boot this first image file as-is with the emulator (see section 5), the emulation crashes on loading the system extensions. We solved this problem [38] by creating another disk containing a fresh installation of the vintage operating system, from which we could access the first disk image and disable the system extensions in question (“Video Startup” and “A/ROSE”).

#### 4.2 Problems in the Run Time Environment

Tests on the Performa computer uncovered a reproducible error that leads to a crash of the HyperCard application: When invoking the audio playback on certain text cards (33-42), the application displays an error message “Unerwarteter Fehler 5454” [unexpected error 5454], forcing the user to quit the application,

see Figure 4. After re-launching the application, the buttons for selecting the individual cards and for playing audio do not react until the user either hits the “stop audio” button or navigates to the Hypertext’s reference layers, and back again.

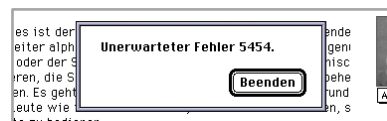


Figure 4. Unexpected error

While the previous error already occurred on the original computer, the following problem could only be observed when copying and running the Flusser Hypertext files on other machines (even of the same model series), or when running them in the emulator: The Hypertext executes, but fails to output any sound at all. The audio function of the HyperCard framework as such was tested, and it worked on all systems. All

<sup>3</sup> Incomplete supplementary cards of the development version had subsequently been filled with content by Bernd Wingert (between 1994 and 2007). No cards were added, and the structure was left unchanged.

<sup>4</sup> image taken from the Vilém Flusser Archive collection (© Vilém Flusser Archive)

attempts to solve this problem by altering the run time environment failed. The original machine seemed to remain the only system capable of playing the Hypertext's audio content.

Online sources indicate that error 5454 relates to corrupted HyperCard files. We traced the defect back to a particular file ("Ton 8") that holds the audio data of the affected text cards. This file produced the same error when opened in HyperCard directly.

We found a way to inspect the Hypertext's source code (written in HyperCard's scripting language HyperTalk). We traced function calls that relate to audio, and found some functions ("FSound", "Search", and "Fplay") that were neither defined in the code, nor part of the HyperTalk language. Instead, these functions are contained in precompiled libraries which get stored along with the Flusser files, see Figure 5. We also found a special driver library (".SndDriver") that might possibly be involved in the error.

XFCNs from Flusser		
ID	Size	Name
303	970	"SndList"
305	528	"FSound"
530	1488	"Search"

XCMDs from Flusser		
ID	Size	Name
400	2198	"FPlay"

DRVRs from Flusser		
ID	Size	Name
128	13938	Driver: ".SndDriver"

Figure 5. Precompiled libraries and drivers

Apparently, we had encountered a highly intricate problem with the specific run time environment in the form of these libraries.

### 4.3 Patching the Artifact

A reverse engineering of the precompiled libraries was beyond the scope of the project, both technically and with respect to time and effort. Thus, we made the decision to avoid all calls to unknown functions in the Hypertext code, whilst keeping its functionality unaltered.

To eliminate the need for custom audio-related code, we ported [38] the Flusser Hypertext from HyperCard 2.0 to version 2.3, whose "ADDmotionII" standard library provided the desired audio functionality. By replacing the artifact's special audio functions with the standard library functions, we succeeded in enabling audio playback in the emulation as well as on our vintage Mac computers.

Having access to the artifact's code also enabled us to solve the error 5454 issue: In HyperCard, we built a replacement file for the seemingly corrupted "Ton 8" file from scratch, using the original audio resources, which we had salvaged from the "Ton 8" file.

As a result of the whole process we obtained a virtual disk that boots with the emulator and is fully functional. The patched artifact shows no unexpected error messages or even crashes. Otherwise, it appears almost unaltered to the user. In particular, the patches are not obvious to the user as an amendment to the artifact. The only minor exception is a short interruption of the playback of the audio recording (see section 3) in the middle of

each text card, when HyperCard loads the second half of the audio data for the card. This deviation results from the patches that we applied to enable audio playback, and could not be avoided.

The decision to accept such slight deviations was reached by the exhibition curators who were aware of the conservational implications of the technical patches. Yet, a guiding theme of the exhibition was to reveal the continuing effect of Flusser's philosophical writings on current phenomena in arts and media. An early, striking example for such a crosslink was the Flusser Hypertext, an innovative "multimedia study system" that reflected Flusser's ideas (see section 3). Hence, the curators favored a smooth user experience and the stability of the emulation during the exhibition over absolute fidelity to the inherited object at the code level.

The patches are marked by comments in the code, but cannot be reversed at the push of a button. Certainly, we saved different intermediate versions of the code during the patching process, including a completely unaltered disk image which can serve as the starting point for more traditional conservation approaches.

## 5. EMULATION

The vintage Macintosh computer that contained the snapshot of the Flusser Hypertext features a Motorola 68040 CPU running System 7.1.2 as its operating system. This type of computer can be emulated using the "BasiliskII" emulator that is freely available for Windows, Linux, and OSX hosts. The Flusser Hypertext poses no special hardware or interface requirements to the emulator.

At the beginning of the preservation project, we had little experience with this particular emulator. We asked colleagues from the University of Freiburg for support, who kindly helped by setting up the initial emulation.

### 5.1 Emulation during Preparation

Once we had managed to produce a virtual disk that was bootable with the emulator (see subsection 4.1), we made extensive use of emulation during the preparation phase. We found this to be a convenient approach since our artifact required an unusually long trial and error-phase of changes to its run time environment and even code (see subsection 4.2). If some change failed, it was much easier to return to the last working copy of the virtual disk under emulation than to restore the previous state of the real hard disk in the vintage computer.

We used a stand-alone setup of the BasiliskII emulator during preparation. A stand-alone setup must be manually configured, but provides fast emulator start-ups, direct access to the emulator's configuration parameters, and easy restoration of any input file that did not work as desired or got damaged in a trial run, by simply overwriting the file with a backup copy.

Before making any changes permanent, we tested on our vintage computers whether the changes had any unwanted impact on the behavior of the artifact.

### 5.2 Configuring the Emulation

For the actual exhibition, the BasiliskII emulator was packaged into a stand-alone, bootable-from-stick version of the Freiburg emulation framework [12].

The bootable stick comprises of a special boot loader and two partitions. One partition holds the Linux host system, the emulation framework, and the emulator. The other partition holds the virtual disk file with the patched Flusser Hypertext files, a Mac Quadra 630 ROM image, and the configuration data for the emulator.

Using this pre-fabricated stick had practical advantages for us: The Linux host system boots automatically when power is

turned on at the mini PC that served as the host computer in the exhibition. The Linux system also was configured to automatically start the emulator at boot time with (a fresh copy of) the virtual disk. In addition, some keyboard shortcuts were disabled to prevent users from gaining access to the underlying Linux host system.

The screen resolution in the emulator was set to 640x480 pixels, a resolution which is typical for Mac computers of the time. The color depth was set to maximum. Otherwise, standard values were used for the configuration parameters of the emulator (see subsection 5.5).

### 5.3 Sound Problem

Despite its benefits, the automated framework approach turned out to have significant drawbacks: Since the emulator was completely encapsulated within the framework, which booted immediately to the emulation when turning power on, there was no direct access to the configuration of the emulator, nor to the virtual disk.

This caused problems right before the exhibition started, when we discovered that the sound was missing in the emulation.

The sound must be activated at all levels of the emulation: in the host system, in the configuration of the emulator, in the options (if any) of the artifact, in the emulated Mac system, and, finally, at the speakers. The problems were resolved in Freiburg.

For future exhibitions, we are seriously considering to use a stand-alone emulator in the exhibition instead of a packaged emulator, providing for more direct, in-house control over the emulator configuration.

### 5.4 Peripheral Devices

The screen resolution for the emulation is 640x480 pixels. Presenting such a low resolution on a modern display results in a small area on screen; alternatively, scaling this up to the size of a modern display results in a blurred or even distorted picture. In addition, the curatorial goal was to preserve as much of the original look-and-feel of the Flusser Hypertext as possible in the emulation.

Hence, the curatorial decision was made to use original, resp., true to original peripheral devices in the exhibition.

We used a 15 inch vintage Apple multiscan color display. This particular model is slightly more recent (1994-96) than the Flusser artifact, but it comes with a VGA port instead of the more typical Apple DB-15 port, which made it easy to connect the display to the mini PC using a standard VGA cable. No special adapter was needed, as opposed to 14 inch Apple displays.

We also used a vintage Apple extended keyboard and Apple mouse. To connect them to the mini PC, an Apple Desktop Bus (ADB) to USB adapter was required. Such adapters are still available over the Internet for a reasonable price.

For the sound output, we used a pair of external vintage speakers placed next to the display and connected with standard audio cables to the mini PC.

Only the display, keyboard, mouse, and speakers were placed on top of the table used in the exhibition — the mini PC was hidden underneath the table (see Figure 6) and was not easily visible to the visitors. The whole arrangement looked quite authentic; as if it came straight from the 90s, see Figure 7.



Figure 6. Mini PC under the table<sup>5</sup>



Figure 7. Emulation in the exhibition<sup>6</sup>

### 5.5 Memory Configuration

In an emulation, the main memory (RAM) required by the vintage operating system for its execution gets emulated; that is, it is provided by the emulator program through software means. The amount of main memory to be emulated must be configured as a parameter of the emulator.

When setting this parameter to a common value of 32 MB (the Performa 630 has a physical maximum of 36 MB), our virtual disk booted under emulation, but the Hypertext failed to start, complaining about not having enough memory, see Figure 8.



Figure 8. Not enough memory

When an application requires more memory than is available as physical RAM on a vintage Mac, the administrator can reserve part of the hard disk as so-called virtual memory, which then is added automatically by the Mac operating system to the total memory available to applications.

In the BasiliskII emulator, support for virtual memory is not implemented, though: When activating virtual memory in the emulated Mac, the emulator crashes. The quick solution is to simply set the amount of *emulated* RAM to the desired total value. For the Flusser Hypertext, setting the emulator's memory parameter to 128 MB works; we used this value in the exhibition.

At a later occasion, though, we observed that only a small fraction of the reserved memory actually gets consumed by the HyperCard application that processes the Flusser Hypertext, see Figure 9.

<sup>5</sup> image © Martin Häberle

<sup>6</sup> image taken from [https://www.itas.kit.edu/2015\\_032.php](https://www.itas.kit.edu/2015_032.php)  
(© Karlsruhe Institute of Technology)



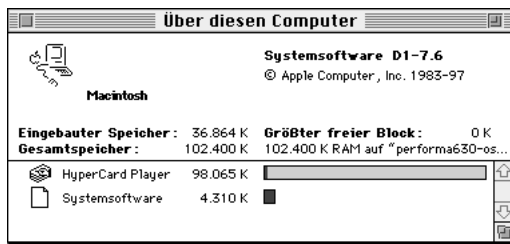


Figure 9. Actual memory consumption

On a vintage Mac system, the amount of memory that a program claims at start-up is preset and can be viewed in its “Information” window. For the HyperCard program, this value was set to 60,000 KB, which is far more than is actually needed for processing the Flusser files. Changing this value to more moderate 20,000 KB is completely sufficient and eliminates the need to use virtual memory on the vintage Macs, resp., to use an exceedingly high value for the emulator’s memory parameter.

This issue demonstrates that settings in the emulated operating system can feed back into the configuration of the emulator, and vice versa.

## 6. OPERATION

The emulation operated continuously for more than 4 months in the exhibition. During this period, we made a number of instructive observations concerning the proper setup and operation of emulations in a museum and exhibition setting. In some cases, we learned that we should better do certain things differently in future emulation-based exhibitions.

We also extended our comparison of the look-and-feel of the Flusser Hypertext under emulation against its original appearance when executing on restored, vintage Mac computers. This allowed us to better assess the quality of the emulation, and it provided additional observations about the run time environment required by the artifact.

### 6.1 Deleting the Object

After booting-up the host system, the emulation automatically starts (see subsection 5.1). The emulator presents the desktop of the vintage Mac system to the user, including an icon in the center for starting the Flusser Hypertext (see Figure 10), waiting to get double-clicked. The desktop also offers two supplementary videos that explain the usage of the Flusser Hypertext; the videos were produced in the 90s along with the Flusser project.

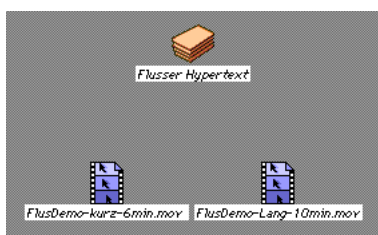


Figure 10. Desktop icons for the Hypertext and videos

Clearly, this is *not* a fail-safe setup for a public exhibition. Instead of starting the digital artifact, the user can access various Mac system functions using the menu bar at the top edge of the desktop. The user can even *delete* the object by dragging it into the waste basket, see Figure 11. This will leave the emulation in an unusable state for the next visitor, who will probably be confused and turn away from the object without telling anybody.

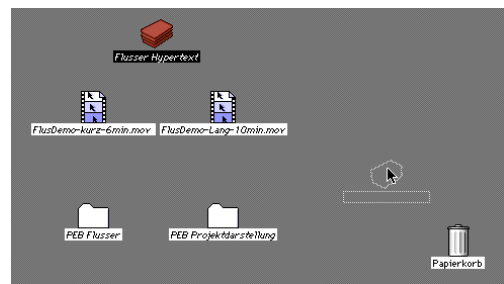


Figure 11. Dragging the artifact into the waste basket

When the object has been deleted, a functional state can only be recovered by re-booting the emulation. This problem actually occurred during the Karlsruhe exhibition. In the sequel, the technical staff kept an eye on the Hypertext exhibit to take corrective action if necessary. For the exhibition in Berlin, the supervisory staff was asked to check the state of the emulation several times a day; if something was wrong, they re-booted the whole system.

Our approach was motivated in part by the desire to make the videos accessible to the visitors, but we might better have presented the videos on a separate computer and display.

For future exhibitions, a more elaborate and fail-safe approach is needed. In a museum setting, not only the emulator should start automatically, but even more so the digital object itself. Users should not be allowed to exit the running object. If the exit function cannot be de-activated within the object itself, the object must get restarted automatically and immediately by the system whenever a user chooses to quit.

This automatic restart-feature for the object must be implemented at the level of the emulated operating system, combining autostart features of the vintage system with a special “watchdog” program that runs in the background to supervise the object in question: If the object (more precisely, its application process) stops running for some reason, the “watchdog” restarts the object.

For some digital objects, it may also be appropriate to restart the object whenever it has been inactive for some time, because visitors typically just leave one exhibit and move to the next. An automatic reset will always present a tidy object to the next visitor. Inactivity can be detected using a watchdog programmed with a time-out.

### 6.2 Exit

After the emulation has started, the desktop of the vintage Mac system is presented to the user. Similar to the problem of deleting the digital object as described in the previous subsection, a user can also deliberately shut down the emulated Mac system using the menu bar on the Mac desktop. This occasionally occurred in the Karlsruhe exhibition. When the emulated Mac shuts down, the emulator program itself exits automatically, and control returns to the underlying host system. The emulation framework then presents a screen that asks the user to restart the emulation.

Again, this is *not* a fail-safe setup for a public exhibition. There should be no way for museum visitors to exit the emulation or shut down the emulated system. If the shutdown functionality cannot be de-activated at the level of the vintage system, the emulation must restart immediately without manual intervention. This can be achieved using standard start-up and process control features of the underlying host operating system.

### 6.3 Pixel Errors

In the emulation, some spots occur in the small Flusser photograph that is placed at the right hand side of the text cards (see Figure 12). At first, we thought that these spots were pixel errors introduced somehow by the emulation. Yet, a quick comparison revealed that the spots are also visible when running the patched artifact on a vintage Mac; that is, the pixel errors were nothing but false positives.



Figure 12. Pixel errors

This issue serves as an example that original hardware not only is an indispensable tool to identify deviations of the emulation from the original, but, contrary, also helps to confirm that the emulation actually *conforms* to the original appearance.

### 6.4 Missing Fonts

A problem that we often encounter in emulations of digital objects is missing fonts. For the Flusser Hypertext, this initially seemed to be no issue, since we used a hard disk image of the vintage computer as the basis for the virtual disk in the emulation, and that image contained all necessary add-ons, including fonts. But when installing the Flusser data and code files on our restored vintage Macs, which carried a fresh system installation, the missing font problem popped up: As the vintage operating system substituted a standard font for some missing fonts automatically, the text on each card was incomplete and the layout looked somewhat distorted.

The problem would also have occurred if we would have taken the (quite common) approach of using two virtual hard disks for the emulation instead of one, separating the operating system to be emulated from the files of the digital object.

Under the Mac operating system, required fonts must be copied to the proper system folder. Our disk image contained several dozens of fonts that had been added to the development system after installation; hence, determining the exact set of required fonts was quite time-consuming. The Flusser Hypertext actually requires 3 non-standard fonts.

This example shows that original hardware is often helpful to identify additional, hidden dependencies of the digital artifact on its run time environment.

### 6.5 Pace of Operation

We conducted a few experiments to evaluate whether the emulation shows any noticeable difference in its pace of operation as compared to the Flusser Hypertext running on a vintage Mac computer. The emulator executed on a standard laptop (dual-core at 1.9 GHz, 4 GB of memory, Windows 7); the vintage Mac was a restored Performa 630 (68040 at 20 MHz, 36 MB of RAM, no virtual memory, Mac system 7.6).

The Flusser Hypertext is largely static in nature, changing screen only when the user interacts with the program; typically, when moving to the next card by clicking on a number at the bottom or right edge of the current card. The only “animated” behavior can be observed when the recording of the lecture is played. In this mode, the system automatically moves from card to card, in sync with the recording, see section 3.

There is no noticeable speed difference when listening to any individual card. Overall, the emulation is slightly faster than the original Mac, lying ahead by 1.5 text cards at the end of the whole lecture, which consists of 49 text cards. On the vintage

Mac, there is a slightly longer delay when the system moves from card to card. The emulator running on modern hardware seems to be faster when it comes to loading the next data into the HyperCard program.

The total gap is much larger when comparing the Performa 630 (no virtual memory) against another vintage Mac (Quadra 650, 68040 at 33 MHz, 36 MB of RAM, Mac system 7.6) that has virtual memory enabled. Although the Quadra is the faster computer, the virtual memory mechanism slows down the loading of data significantly. The Performa with no virtual memory is ahead by 1 card already after about 1/3 of the lecture.

The experiments show that the audio sequences are replayed faithfully in the emulation, which is a key factor for the authenticity of any emulation. In addition, we learned that variation in the vintage hardware or configuration can have a much larger impact on the appearance of a digital object than the emulation. Such measurements are impossible to conduct without having original hardware at hand.

### 6.6 Hanging Print Function

Except for the text cards containing the transcript of Flusser’s lecture, all cards in the Flusser Hypertext can be printed out using an icon in the lower right corner of each card, see Figure 13. This includes the cards containing supplementary material for Flusser’s lecture, such as explanatory articles or bibliographies, and the personal note cards created by the user (see section 3).

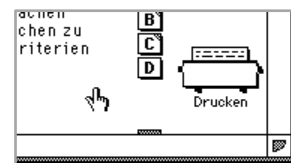


Figure 13. Print function

In the exhibition, no printer was connected to the host computer, and the emulation was not configured to accept and handle print requests. Nonetheless, it was possible for a visitor to click on the printer icon and create print jobs. If done repeatedly, a long print queue emerged that blocked the user interface – the program became irresponsive and seemed to “hang.”

The Flusser Hypertext does not offer an option for deactivating its print function similar to deactivating its audio function in the settings. Hence, either a real printer must be added and the emulation configured accordingly, or, a non-blocking “mock printer” must be installed at the level of the emulated vintage Mac system. The problem was discovered only late in the Karlsruhe exhibition. For the Berlin exhibition, we left the virtual disk as is and relied on our supervisory staff to handle any problems.

This issue highlights the fact that digital objects can include features that must be explicitly handled at the technical level in an emulation. For objects with a complex internal structure, such features need not be as obvious as the print function, but can be buried rather deep inside the object. F.e., in other objects under preparation we encountered hidden links to the internet. Such features might even be undocumented, especially in digital art that often includes elements of surprise which the user is expected to discover when interactively exploring the piece of art.

## 7. CONCLUSIONS

In this paper, we presented a real case study that illustrates the whole process of preserving a digital object in the museum by means of emulation. The required steps range from creating the virtual disk to operating the emulation in a public exhibition.

We encountered a number of instructive technical and curatorial problems that resulted in a number of specific lessons that we learned, some typical for any emulation-based preservation, others typical for the exhibition setting that we worked in.

*(L1) Preparing an artifact for the emulation demands a close analysis of the run time environment needed by the artifact.*

This includes the identification of special drivers, non-standard support files, external interfaces, and special system settings (cf. subsections 4.2, 5.3, 6.4, 6.6).

The ties of a digital object into its run time environment can be very subtle; one missing detail can lead to a strange behavior or failure. In our experience, the required run time environment often is not documented in sufficient detail even for artifacts published on distributable media, turning the analysis into a trial-and-error process.

*(L2) Emulators typically do not support every feature of the vintage system.*

Under emulation, drivers may fail to work or crash the emulator, system options may crash the emulator when activated, or external interfaces may be unavailable. This creates problems when trying to install and/or execute an object with its run time environment under emulation (4.1, 4.2, 5.5).

In practice, it is not always possible to pin down and fix the root-cause of a vintage system-level problem. It may be necessary to circumvent the problem by de-activating system features, exchanging certain drivers, or moving to another version of the vintage application program and operating system. Patching the digital object (as we did) should be a last resort.

*(L3) Emulating a whole system-image can be a non-trivial task.*

A full image might reflect an intermediate version of a digital object (as in our case), or conserve a complete heritage work environment. Assuming a non-networked system, the image will contain all required run-time components; yet, some components may not work under emulation (4.1, 4.2).

The complexity of a full image of a “living” system exacerbates the difficulties of fixing inconsistencies in the run time environment under emulation. When the artifact to be emulated is not finalized, tested software, it may contain bugs, and its documentation will likely be fragmentary or missing, adding to the problem.

*(L4) In the museum, curatorial judgments provide the direction for the technical setup of the emulation.*

Curatorial judgments refer to the choice of peripheral devices, the acceptance or rejection of deviations of the emulation from the original, the user interface offered, and the acceptance of any changes to the artifact (5.4, 6.5, 6.1, 4.3).

An emulation need not necessarily be perfect. Augmenting the emulation with true to original peripheral devices can provide a way to preserve the essentials of the original look-and-feel.

In our case, even patches to the artifact seemed admissible from a curatorial perspective, given two facts: The Flusser Hypertext was not a final, production-quality release, but an advanced prototype; the computer on which it was stored was not Flusser’s own personal computer, hence, it made little sense to

try and preserve something like Flusser’s “digital working environment.”

*(L5) In a museum exhibition, the emulation must be specifically safe-guarded.*

In a public exhibition, the continuous operation of the emulation must be guaranteed. This poses technical challenges for the setup of the emulation, in excess of configuring the emulator. The artifact itself must be protected against deletion, and the emulation must be sealed to prevent any unintended usage of the emulator, vintage system, or underlying host system (6.1, 6.2, 6.6).

Visitors should best be prevented from quitting the running artifact at all. This requires elaborate technical measures at the vintage and host system level.

*(L6) Preparing an emulation requires substantial technical knowledge and skills.*

Knowing the configuration options of the emulator certainly is a prerequisite to tailor the emulation to the object (5.2, 5.5). In addition, in the preparation phase frequent trial runs of the emulation are typical, until a working overall setup is found (5.1, 5.3, 5.5). To achieve short cycle times, a stand-alone emulator is better suited than a framework, which encapsulates the emulator and input files into additional software layers and special workflows. Yet, such a stand-alone setup must be manually configured.

Knowledge of the emulator is not sufficient to prepare the one central input for the emulation: the virtual disk. The virtual disk contains the vintage run time environment of the object to be emulated. An understanding of and practical experience with the vintage operating system, its hardware and application programs is indispensable for solving any problems of incompatibility of the digital object with the emulator, by customizing the virtual disk (cf. L2). This also applies to safe-guarding the emulation in an exhibition (cf. L5). In addition, settings in the vintage system can feed back into the configuration of the emulator, and vice versa (5.5, 5.3).

*(L7) The preparation of a virtual disk for the emulation consumes the lion’s share of the total effort.*

The virtual disk contains the run time environment for the emulated artifact, compensates any shortcomings of the emulator, and reflects the special technical measures taken for an exhibition setup. The more complex the digital object, the larger the preparation effort. Even for more average objects of digital art, which do not require analyzing and patching the artifact’s code as in our case, we found that the effort for preparing the virtual disk (including the time-consuming trial runs of the emulation during preparation) typically is substantial and far exceeds the effort for installing and configuring the emulator. This fact seems to get severely underestimated in the literature.

*(L8) Original hardware is a valuable, sometimes indispensable tool in the preservation process.*

Comparing the emulation against the original is obligatory for assessing the quality of the emulation in an art context. Comparing is easier when the original hardware is still working (6.3, 6.5). An original hardware environment is also a valuable tool when tracing problems in the emulated run time environment (4.2, 5.5, 6.4).

Hence, museums should start their preservation effort while the hardware for their digital artifact is still functional and spare parts are still available.



Our digital object under preservation, the Flusser Hypertext, is an important cultural artifact that was already close to getting lost forever: only the binary files of this work still existed, stored on the physical hard disk of a vintage computer. We now have a salvaged version on hand, in the form of a disk image file that can be easily copied to various media, distributed, and placed into long-term digital storage. This version of the Flusser Hypertext is ready to execute using an “off-the-shelf” emulator.

## 8. ACKNOWLEDGMENTS

Thanks go to Dirk von Suchodoletz and Klaus Rechert from the University of Freiburg, who helped with the hard disk image and the setup of the emulation, and who provided their emulation framework as a bootable from stick-version. We’d also like to thank Baruch Gottlieb for establishing this contact.

The German Research Foundation DFG supported the first author with a research grant.

Adam Rosen from the Vintage Mac Museum in Boston kindly made us aware of how to overcome access restrictions to the source code of locked HyperCard stacks.

We’d also like to thank Bernd Wingert for donating his Macintosh Performa computer with the Flusser Hypertext to the Vilém Flusser Archive in Berlin, and for freely sharing his inside knowledge of the Flusser Hypertext.

## 9. REFERENCES

- [1] Rothenberg, J. 1995. Ensuring the Longevity of Digital Information. *Scientific American*, 272(1) (Jan. 1995), 42-47
- [2] Holdsworth, D., and Wheatley, P. 2001. Emulation, Preservation and Abstraction. In *Research Libraries Group RLG DigiNews* 5, 4 (Aug. 15, 2001) Online at <http://sw.ccs.bcs.org/CAMiLEON/dh/ep5.html>
- [3] Farquhar, A., and Hockx-Yu, H. 2007. Planets: Integrated Services for Digital Preservation. *Int. Journal of Digital Curation* IJDC 2, 2 (2007), 88-99.
- [4] KEEP project. Project information online at [http://cordis.europa.eu/project/rcn/89496\\_en.pdf](http://cordis.europa.eu/project/rcn/89496_en.pdf)
- [5] Cochrane, E. 2014. Emulation as a Service (EaaS) at Yale University Library. Online at <http://blogs.loc.gov/digitalpreservation/2014/08/emulation-as-a-service-eaas-at-yale-university-library/>
- [6] Brown, G. 2012. Developing Virtual CD-ROM Collections: The Voyager Company Publications. *Int. Journal of Digital Curation* 7, 2 (2012), 3-20.
- [7] Pinchbeck, D., Anderson, D., Delve, J., Otemu, G., Ciuffreda, A., and Lange, A. 2009. Emulation as a strategy for the preservation of games: the KEEP project. In *Proc. of the Int. Conf. on Breaking New Ground: Innovation in Games, Play, Practice and Theory* (Brunel University, London, UK, Sep. 2009), DiGRA 2009.
- [8] Loebel, J.-M. 2014. *Lost in Translation* [in German]. Ph.D. Dissertation. Humboldt University Berlin & VWH Verlag, Glückstadt.
- [9] Matthews, B., Shaon, A., Bicarreguil, J., and Jones, C. 2010. A Framework for Software Preservation. *Int. Journal of Digital Curation* 5, 1 (June 2010), 91-105.
- [10] Braud, M., Lohman, B., and van der Hoeven, J. 2012. How to run emulators remotely via the Emulation Framework. Online at <http://emuframework.sourceforge.net/docs/EF-howto-remotemulation-1.0.pdf>
- [11] Satyanarayanan, M. 2013. Olive: One-click Execution of Internet-Archived Software. In *NYU Scientific Reproducibility Workshop* (New York, USA, May 30, 2013).
- [12] Liebetraut, T., Rechert, K., Valizada, I., Meier, K., and von Suchodoletz, D. 2014. Emulation-as-a-Service: The Past in the Cloud. In *7th Int. Conf. on Cloud Computing CLOUD* 2014. 906-913.
- [13] Rechert, K., Liebetraut, T., Stobbe, O., Valizada, I., and Steinke, T. 2015. Characterization of CDROMs for Emulation-based Access. In *12th Int. Conf. on Digital Preservation* (Chapel Hill, USA, Nov. 2-6, 2015). IPRES 2015.
- [14] Rinehart, R., and Ippolito, J. 2015. *Re-collection: Art, New Media, and Social Memory*. MIT Press.
- [15] Kaufmann, F. 2013. Hacking Mondrian. In *Digital Art Conservation: Theory and Practice in the Conservation of Digital Art*, B. Serexhe (Ed.). Ambra V, Vienna, 273-284.
- [16] Padberg, F. 2014. Emulation of Media Art or Art-Handling in the Change of Technology [in German]. In *Int. Symp. on Art-Handling* (Migros Museum for Contemporary Art, Zurich, Switzerland, Nov. 27-28, 2014).
- [17] Falcao, P., Ashe, A., and Jones, B. 2014. Virtualisation as a Tool for the Conservation of Software-Based Artworks. In *11th Int. Conf. on Digital Preservation* (Melbourne, Australia, Oct. 6-10, 2014). IPRES 2014.
- [18] Fino-Radin, B. 2011. Digital Preservation Practices and the Rhizome Artbase. Technical Report. Online at <http://media.rhizome.org/artbase/documents/Digital-Preservation-Practices-and-the-Rhizome-ArtBase.pdf>
- [19] Espenschied, D., Rechert, K., von Suchodoletz, D., Valizada, I., and Russler, N. 2013. Large-Scale Curation and Presentation of CD-ROM Art. In *10th Int. Conf. on Digital Preservation* (Lisbon, Portugal, Sep. 2-6, 2013). IPRES 2013.
- [20] Lurk, T., Espenschied, D., and Enge, J. 2012. Emulation in the context of digital art and cultural heritage. *Praxis der Informationsverarbeitung und Kommunikation* PIK 35, 4 (2012), 245-254.
- [21] Lorrain, E. 2013. PACKED Case Study report: Mondophrenetic (2000, Herman Asselberghs, Els Opsomer, Rony Vissers). Online at <http://www.scart.be/?q=en/content/case-study-report-mondophrenetic%E2%84%A2-2000-herman-asselberghs-els-opsomer-ronyvissers-0>
- [22] Besser, H. 2001. Longevity of Electronic Art. In *Int. Cultural Heritage Informatics Meeting* (Milano, Italy, Sep. 3-7, 2001). ICHIM 01.
- [23] Irrgang, D., and Marburger, M. R. 2015. Vilém Flusser – A Biography. In *Flusseriana. An Intellectual Toolbox*, S. Zielinski, P. Weibel, and D. Irrgang (Eds.). Univocal Publishing, Minneapolis, 452-519.
- [24] Böhle, K., Riehm, U. and Wingert, B. 1997. *Vom allmählichen Verfertigen elektronischer Bücher. Ein Erfahrungsbericht* [in German]. Campus Verlag, Frankfurt/M. and New York.
- [25] Gottlieb, B. 2015. Hypertext. In *Flusseriana. An Intellectual Toolbox*, S. Zielinski, P. Weibel, and D. Irrgang (Eds.). Univocal Publishing, Minneapolis, 212-214.
- [26] Flusser, V. 1985. *Into the Universe of Technical Images*, N. A. Roth (Transl. 2011). University of Minnesota Press.

- [27] Wingert, B. 1992. Schreiben für Publizieren. Ein Hypertext-Experiment mit einem Flusser-Text [in German]. In *Kunstforum International* 117 (1992), 109-110.
- [28] Wingert, B. 1991. Erfahrungen bei der Entwicklung eines Hypertextes [in German]. (talk and prototype demo). In *Arbeitskreis Literatur im Informationszeitalter* [Workshop on Literature in the Information Age] (Wissenschaftszentrum Nordrhein-Westfalen, Düsseldorf, Germany, June 20, 1991).
- [29] Wingert, B. 1996. Kann man Hypertexte lesen? [in German]. In *Literatur im Informationszeitalter* [Literature in the Information Age], D. Matejovski and F. Kittler (Eds.). Campus Verlag, Frankfurt/M. and New York, 184-218.
- [30] Wingert, B., 1991. (prototype demo). In *CULTEC – Kultur und Technik im 21. Jahrhundert* [Culture and Technology in the 21st Century] (Wissenschaftszentrum Nordrhein-Westfalen, Essen, Germany, Nov. 22-23, 1991).
- [31] Böhle, K., Riehm, U, and Wingert, B. 1992. (prototype demo). In *Workshop Hypersystem-Konzepte in Medien und kultureller Produktion II* [Workshop on Hyper-system Concepts in the Media and Cultural Production II] (University of Lüneburg, Germany, July 13-15, 1992). Workshop program and list of exhibits online at <http://www2.leuphana.de/hyperkult/archiv/hk2.pdf>
- [32] Wingert, B. 1992. Flusser-Hypertext. Prototyp und Entwicklungserfahrungen [in German]. (talk and prototype demo). In *GI-Symposium Hypertext und Multimedia. Neue Wege der computerunterstützten Aus- und Weiterbildung* [Symposium of the GI on Hypertext and Multimedia. Novel Approaches in Computer-supported Education], U. Glowalla and E. Schoop (Eds.). Springer, Berlin, 137-144 and 356. (Schloss Rauischholzhausen/Marburg, Apr. 28-30, 1992).
- [33] Wingert, B., and Riehm, U. 1992. Wie wirken Hypertexte? [in German]. (talk and prototype demo). In *Ergebnisse der 12. Arbeitstagung Mensch-Maschine-Kommunikation* [Results of the 12th Workshop on Man-Machine Communication], S. Dutke (Ed.). Free University of Berlin, Institute of Psychology, 41-50. (Berlin, Nov. 15-18, 1992).
- [34] Wingert, B. 1993. Die neue Lust am Lesen? Überlegungen zur Lesbarkeit von Hypertexten [in German]. (talk and prototype demo). In *2nd Int. Vilém-Flusser-Symposium* (Antwerpen, Belgium, Oct. 28-31, 1993). List of talks and exhibitors online at <http://www.flusser-archive.org> (▷ Vilém Flusser ▷ Symposien und Konferenzen zu Flusser ▷ 1993: Zweites Internationales Vilém-Flusser-Symposium)
- [35] Wingert, B. 1995. Die neue Lust am Lesen? Erfahrungen und Überlegungen zur Lesbarkeit von Hypertexten [in German]. In *Kursbuch Neue Medien. Trends in Wirtschaft und Politik, Wissenschaft und Kultur* [New Media Guide. Trends in Economy and Politics, Science and Culture], S. Bollmann (Ed.). Bollmann Verlag, Mannheim, 112-129.
- [36] Wiechens, P. 1998. Hypertext und Künstlerbuch. Das Buch nach dem Ende des Buches [in German]. In *Einführung in die Kulturwissenschaft* [Introduction to Cultural Studies], T. Düllo, J. Greis, C. Berthold, and P. Wiechens (Eds.). LIT Verlag, Münster, 328-346.
- [37] Flusser, V. 1991. Hypertext. Über das Schicksal von Büchern [in German]. In *NZZ Folio* 10 (Oct. 1999), 35-36. Online at <http://folio.nzz.ch/1991/oktober/hypertext>
- [38] Tögel, P. 2016. *Denk-Maschinen – Flussers Digitale Publikationen in der Ausstellung “Bodenlos – Vilém Flusser und die Künste”* [in German]. Master Thesis. Berlin University of the Arts.