Web browsers prescribe the ways we access and navigate knowledge and communities online. Since the 1990s browser software has been an arena for artistic interventions ranging from quirky standalone browsers to performative pieces to minimalist browser add-ons. The (im)possibility of navigation is not taken for granted and is probed, questioned, and reformulated through such software practices. We propose navigation as a mode of exploring interactive software that allows researchers to collectively document manifold facets of artists' browsers.

Navigation

Reihe
Begriffe des
digitalen Bildes

Navigation

Edited by
Inge Hinterwaldner
Daniela Hönigsberg
Konstantin Mitrokhov

Munich, 2022
Open Publishing LMU

Content

Daniela Hönigsberg

Documenting %WRONG Browser .co.kr

There can be no finite procedure that is guaranteed to find all the regularities of an entity.[1]

Conception and preparation

The preparation of the documentation for the *.co.kr* browser of JODI's *%WRONG Browser* series was conducted in an attempt to create a method that would be applicable for all the artistic browsers that are part of our research project. The first step therefore was to contemplate how to capture an artwork that is not only dynamic and responsive to user interaction, but also connected to a complex environment. The central points I determined as relevant for a thorough documentation are as follows:

- The programmed reality of the software, its settings, behaviour and the underlying rules
- The execution of the application, its behaviour and possible user interaction
- The (historical) technical and visual program environment
- The user experience and user behaviour.

The first considerations in this process were of a technical nature: the hardware and software required to run the application, and the tools to create a recording of its audio-visual output. I considered that the optimum way to document the work would be to approximate the original situation in which

[1] Murray Gell-Mann: Regularities and Randomness. Evolving Schemata in Science and the Arts. In: John Casti & Anders Karlqvist (eds.): Art and Complexity. Amsterdam 2003, pp.47-58, here p.50.

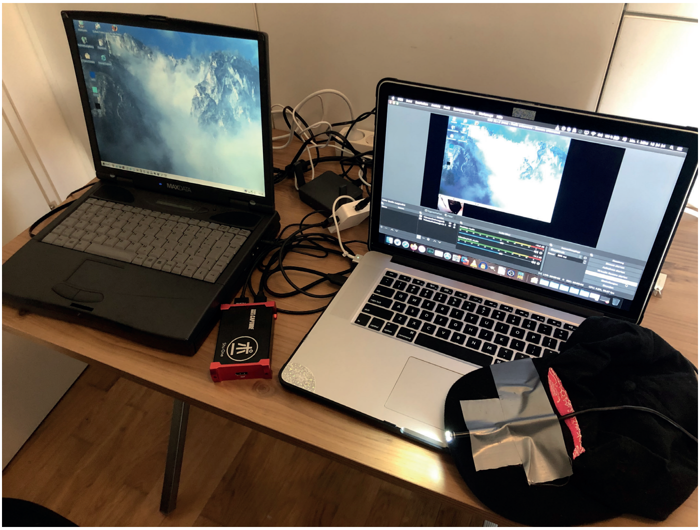
the software was executed. This is what I attempted.[2] The greatest obstacle to this approach is certainly the Internet itself as it has changed considerably in the last twenty years. It represents an extended and highly complex application environment that is virtually impossible to reconstruct – and certainly not in the scope of this documentational experiment.[3] Considering the dynamic nature of the work, the most suitable approach seemed to be to create a video recording of its usage and the moving images presented on the screen. The recording would also include not only the sounds produced by the work itself, but also those resulting from the user's interaction with the computer, such as clicking or typing, as part of the user's experience of the work. It was also necessary to have a recording of me as the user interacting with the program, so a secondary camera was installed in close proximity to my eyes on a baseball cap. Both signal sources, the computer outputs, including system and surrounding sounds and the secondary camera were recorded simultaneously.[4] This low-tech approach (Fig. 1) did not yield the desired outcome because, most of the time, my movement prevented the secondary camera from recording the typing or mouse interaction, and even prevented me from performing natural movements. The secondary video was therefore sub-par and not very helpful in this specific setup. It also created a bigger issue, as predicted, for the informational value of the recorded session. Central aspects, such as the distinction between specific program behaviours that are the result of automated processes and those that are due to interactions with users, were difficult to reconstruct from just the screen capture and thus lost to the documentation and the analysis.

[2] I decided to use a computer originating at about the same time as the artwork: MaxData VMX, Modell NB 2000-line Eton Pro 14.1" TFT. The operating system was Windows Millennium (ME) Version 4.90.3000.

[3] I contemplated finding a way to limit the bandwidth for example, but there are so many factors to consider, that attempting to revive the internet of 20 years ago would be very difficult and require an elaborate approach. However, this would certainly be an interesting project that would benefit a thorough documentation of historic Internet-based artworks.

[4] I used the open broadcast software OBS to record the video on a MacBook Pro with macOS Mojave Version 10.14.6 via a Ba30DEllylelly Mini VGA to HDMI Converter VGA2HDMI Adapter, a Mira Video Capture Box and a Materro Wi-Fi Endoscope Camera Model YPC99-5.

Fig.1, Technical setup for the recording of the documentation video: MaxData VMX, Model NB 2000-line Eton Pro 14.1" TFT, Windows Millennium (ME) Version 4.90.3000, GL iNet Slate Dual-band Mini VPN Router|Portable Usage for Travel, Home, and Business (GL-AR750S-Ext) connects the XIRCOM RealPort CardBus Ethernet 10/100+ Modem 56 RBEM56G-100 to the wi-fi, Ba30DEllylelly Mini VGA to HDMI Converter VGA2HDMI Adapter, Mira Video Capture Box, MacBook Pro macOS Mojave, Materro Wi-Fi Endoscope Camera Model YPC99-5.



With the technical specifications decided, the next aspects to consider were how to interact with the browser, and what questions or goals to follow during this interaction. The following list of questions was created with the aim of capturing answers in the documentation:

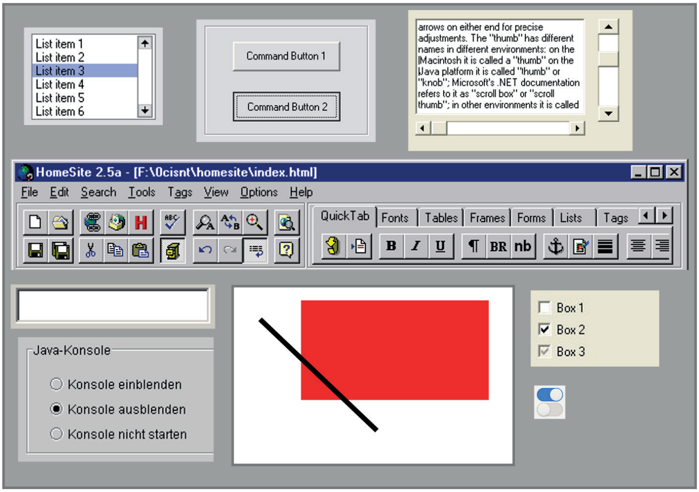
- What does the graphical user interface look like?
- What are the initial settings?
- Does the browser act without user initiative, and what does it do?
- What can the user do?
- What inputs can the user direct?
- What happens after the user directs an input?
- What functions does the graphical user interface offer?
- Is the program's function limited to the browser window?
- Is there an audio output?
- How does the user navigate?
- Are links functional?
- Are there elements or functions familiar from conventional web browsers or browser windows, and which ones?

Initially, the intention was to create a strict sequence of operations for the interaction, so it would be possible to recreate the process and compare the findings. However, as a structured approach was not found, an explorative phase without a strict sequence was included and this became the first part of the documentation.

Exploration phase

The hypothesis used in the exploration phase was that the browser is a graphical user interface (GUI) constructed in a similar way to every other GUI, behaving in a specific way, and containing specific elements that can either be interacted with or not. These specific elements of the GUI have properties that it is possible to discern by looking at and interacting with them. To understand the specificity of the observed GUI, it was consistently compared to a mental aggregational reference

Fig. 2, Aggregational reference model of a GUI with standard graphical control elements: Window, listbox, buttons, scrollbars, menu bar/ toolbar, text box, canvas, check boxes, labels, combo box, toggle switches.



arrows on either end for precise adjustments. The "thumb" has different names in different environments: on the Macintosh it is called a "thumb" on the Java platform it is called "thumb" or "knob"; Microsoft's .NET documentation refers to it as "scroll box" or "scroll thumb"; in other environments it is called

HomeSite 2.5a - [F:\0cisnt\homesite\index.html]

model of a GUI and its potential graphical control elements (Fig. 2). The reasonable assumption that the program is not actually a GUI, is not covered by this approach, which can be problematic because it does not allow for the description of something genuinely new.[5]

The browser was used with the clear goal of finding the aspects of the program that could be identified as discrete elements and functions. Here, the mental reference model was the guide to identifying the different elements, the discrete functions of these elements and their properties by visual inspection over an extended period of time, and by using a multitude of standard behaviours connected to the standard

[5] Cf. James P. Crutchfield: What Lies Between Order and Chaos? In: John Casti & Anders Karlqvist (eds.): Art and Complexity. Amsterdam 2003, pp. 31–45, here p. 41.

graphical control elements. The behaviour of the application however, forced me into taking a rather scattered approach. Especially in the beginning, I jumped from one part of the application to the next purely because something caught my attention, or because the element was no longer active in the automated processes of the program, making it impossible to continue the interaction. Thus, the automated behaviour of this specific application was actively preventing me from following a structured approach and I had to adjust my explorative behaviour accordingly.

The first differentiation between the elements was by the presence or absence of any interactive properties. If there was a behaviour visible that was not due to interaction, then it was clear that this was part of the automated program behaviour and dependent on system internal parameters. Next, I determined aspects such as the approximate size of each element in relation to the other elements. The specific relevance of the size is that when this is known, the consequences of two elements overlapping can be discerned, for example. This provides information about their background settings and similar properties. One obstacle is the lack of information regarding the number of elements there are to observe and with which it is possible to interact. Potentially, there could be, for example, elements that are transparent and do not hold any kind of content but are in motion. These elements might perform a variety of activities, but it would be impossible to identify them, and they could still have an influence on other elements. However, this influence would be very difficult to trace. The automated behaviour in many of the *%WRONG Browser* applications is so complex and heavily randomized that it would be necessary to conduct either a very thorough long-term observation of the program, or an investigation of the source code. The perceived complexity of the executed application is a striking example of a problem well described in the theory of complex systems. "As one moves across the spectrum of predictability – from ordered to random behaviour – the "complexity" is maximized in the middle."[6] And without the source code the regularities governing the behaviour are difficult to discern. Nevertheless, with an investment of considerably more time and resources, including an automated interaction, the method proposed here should reveal these irregularities in such a way that is nearly as effective as consulting the source code of the program. However, without accurate time measurement systems – necessary for example to precisely recreate a specific behaviour – it was not possible to consider or approximate these kinds of eventualities. The remaining option was to perform every conceivable common and uncommon interaction in order to reveal the different properties of each element. Thus, I have attempted to click them, mark them, change their position, and stop them behaving in an automated manner. In addition, I deliberately misused elements whose functions were known or supposed, and tested their limits and hidden features to get to their coded identity.

Some aspects could only be documented by repeating the same behaviour and re-starting the application several times. The only time I used this kind of approach was when I wanted to see if the first automatically inserted URL would always be the same when the browser started up, and which one it was. Even though it was possible to gain much information by pursuing different tasks or different theories in multiple sessions, my focus was to stay with the session for as long as it took me to gain the impression of knowing all aspects that are discernible by a continuous interacting and visual inspection. During the whole recording process, I never exited the appli-

[6] Cf. James P. Crutchfield: What Lies Between Order and Chaos? In: John Casti & Anders Karlqvist (eds.): Art and Complexity. Amsterdam 2003, pp. 31-45, here p. 41., pp. 36-37.

cation and took no notes. This was necessary to ensure that the application's general behaviour was captured in a pure and representative way, as devoid as possible of any outside factors. Staying with the program for a prolonged time was also crucial because nearly all the elements move constantly and erratically, making it difficult to keep track of them.

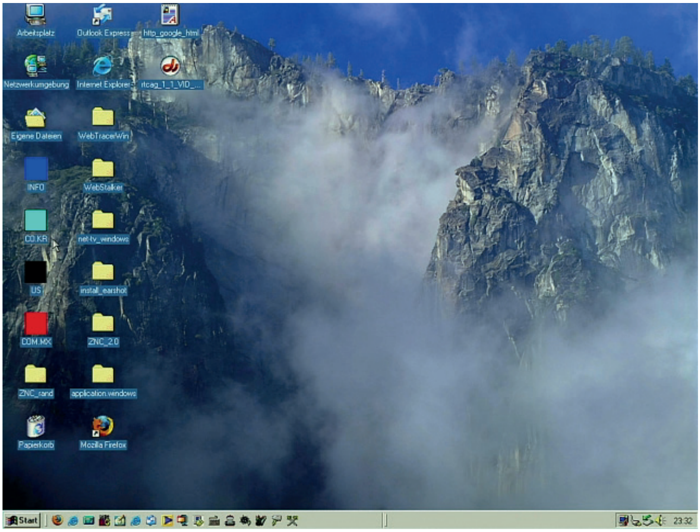
The system environment was another aspect that was included in the recorded documentation as part of the experience of the artwork. The interaction with the browser started with opening the application. Observing the program launch is relevant because it allows us to see the application icon and how it is displayed in the operating system's GUI (Fig. 3), as well as the original setting of the application window, the first of the elements to be compared to the mental reference model. The opening of the *.co.kr* browser into full screen is a property of this element and also has the effect of blocking out the entire operating system's GUI. It would be necessary to leave the application to return to anything else. This is a very immersive approach which evokes a reaction from the user that conventional browsers do not pursue. Viewing the setting of the initial application window seems an important aspect for every browser that is a potential candidate for documentation.

What was not included in the recorded documentation was the retrieval of the executable, or information about how to install it on the system, which would have been an equally interesting aspect to include in the documentation.

Textual documentation and analysis

As described, during the interaction, I used and expanded upon a preconceived mental image of the application, filling a mental model of a GUI with information about the identi-

Fig. 3, Application icon displayed in the operating system's GUI (screenshot): Turquoise square labeled CO.KR.



fied elements and the properties they exhibit. However, this constantly evolving image of what the program is and how it behaves was not written down during the exploration phase. The translation into text was performed in the final step of the documentation, using the approx. 40-minute video of the interaction as the reference.

Here, I was able to describe the program behaviour and elements in the structured way that the exploration phase did not allow. It consisted of:

- the description and images of the technical set up for the documentation itself
- (the reference to the video recording)

- a description and screenshots of the initial screen
- the description and screenshots of the overall structure and composition of the graphical interface, which included the description of each identified element and its properties
- a list of accessed websites and comparative screenshots showing the web pages in *%WRONG browser co.kr* and a conventional browser accompanied by notes of observations.

As the video recording was the only additional source of reference apart from my own memory, the previously mentioned shortcomings of the recording method instantly became evident. The recording of typing activity and the use of the mouse cursor would have been extremely valuable for ascertaining which interactions were performed when, and the actual results of these interactions. The video document created during the exploration phase only allowed me to try and track the mouse cursor on the screen itself. Only the behaviour that was visible on the screen recording could be followed, so the description of the program's behaviour and its properties was primarily based on my mental concept of the browser, and the things that I discerned while using it. The video therefore served to confirm these observations and facilitate a more detailed description. It was useful to look at the video to observe aspects that were not on my mind or did not seem of interest during the interaction. If the video recording had included clear footage of typing activity and mouse usage, it may have been just as helpful as the written documentation.

Conclusion

Understanding what could and could not be done, what was happening and what was possible took longer than expected. Most information was gathered by using the browser over a prolonged period of time, finding an understanding of the application, feeling comfortable with what there was to see and interact with, and resolving the confusion or uncertainty that these programs can create in a user.

A next possible step would be a recorded video presentation of the possible behaviours of the program. This would showcase a very goal-oriented user's behaviour, which is probably not to be considered the natural user behaviour one would expect with this kind of application, as it takes time to understand the program and the behaviour of the elements. Such an approach might lead to an accurate documentation of the possibilities for interaction with the browser and a thorough documentation of its elements. However, what would be lacking is the documentation of the behaviour of the user, which seems to be a crucial part of these works as well. That is, to see what users do and what they need to do, to understand what they see and what the program does. On that note, it must be mentioned that any user who does not intend to create a documentation of the *co.kr* browser, and does not attempt to determine its coded regularities, would be likely to show a significantly different behaviour with the application than I did. The inclusion of numerous participants to document different kinds of user behaviour with the program would therefore be another desirable addition to this method of documentation.

DOI: 10.5282/ubm/epub.93523

Edited by

Inge Hinterwaldner

Daniela Hönigsberg

Konstantin Mitrokhov

Staatliche Hochschule
für Gestaltung Karlsruhe

DFG-Schwerpunktprogramm ‚Das digitale Bild'
Projekt Browserkunst. Navigieren mit Stil

DAS
DIGITALE
BILD

Erstveröffentlichung: 2022
Gestaltung und Satz: Lydia Kähny und Sophie Ramm

Diese Publikation wurde finanziert durch die Deutsche Forschungsgemeinschaft.
München, Open Publishing LMU

DFG Deutsche Forschungsgemeinschaft

UB

DOI 10.5282/ubm/epub.93518
ISBN 978-3-487-16315-4
Library of Congress Control Number
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind abrufbar unter http://dnb.dnb.de

Reihe: Begriffe des digitalen Bildes
Reihenherausgeber
Hubertus Kohle
Hubert Locher

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Deutsches Dokumentationszentrum für Kunstgeschichte Bildarchiv Foto Marburg

Philipps Universität Marburg

Das DFG-Schwerpunktprogramm ‚Das digitale Bild' untersucht von einem multiperspektivischen Standpunkt aus die zentrale Rolle, die dem Bild im komplexen Prozess der Digitalisierung des Wissens zukommt. In einem deutschlandweiten Verbund soll dabei eine neue Theorie und Praxis computerbasierter Bildwelten erarbeitet werden.

9 783487 163154

DAS
DIGITALE
BILD