VCS
VISUAL CULTURE STUDIES

Rivista semestrale di cultura visuale

#3-4 (special double issue)
Art in the Age of Ubiquitous Media
a cura di Sean Cubitt e Valentino Catricalà

MIMESIS

VCS

Tutti i saggi scientifici vengono sottoposti a double-blind peer review.

Per sottoscrivere abbonamenti o richiedere i numeri arretrati scrivere a:
ordini@mimesisedizioni.it
oppure con acquisto online su:
www.mimesisedizioni.it
oppure telefonare al numero:
02-24416383

Abbonamenti e spedizioni

	Italia	Estero
Singolo numero	28,00 €	28,00 €
Abbonamento	50,00 €	50,00€
Spedizione:		
con Poste Italiane	GRATIS	6,00 €
con Corriere	6,00 €	15,00 €

Per procedere all'acquisto è necessario comunicare i seguenti dati: nominativo e codice fiscale; indirizzo di spedizione; ricevuta di esecuzione bonifico da anticipare via e-mail all'indirizzo:
ordini@mimesisedizioni.it

Il pagamento può avvenire tramite: contrassegno; pagamento diretto all'addetto alla consegna (solo per l'Italia e solo per consegna con corriere); circuito paypal all'indirizzo:
ordini@mimesisedizioni.it
bonifico bancario alle seguenti coordinate:
CASSA DI RISPARMIO DI ASTI
Ag. di Sesto San Giovanni
BIC/SWIFT: CASRIT22
IBAN IT 94 T 06085 20700 000000020093
intestato a MIM EDIZIONI srl – Via Monfalcone, 17/19, 20099 Sesto San Giovanni (MI)

MIMESIS EDIZIONI (Milano – Udine)

www.mimesisedizioni.it
www.mimesisjournals.com

Issn: 2724-2307
Isbn: 9788857595900

Via Monfalcone, 17/19 – 20099
Sesto San Giovanni (MI)
Phone: +39 02 24861657 / 24416383
mimesis@mimesisedizioni.it

Indice

Essays

N.B: l'apparato iconografico dei diversi saggi è fruibile presso il sito www.vcsmimesis.org

Essays

Negotiating the way to the Internet
The impact of software design on the browsing experience and user interaction

Inge Hinterwaldner, Daniela Hönigsberg, Konstantin Mitrokhov, Martina Richter

Abstract

During the last 30 years, a variety of Web browsers have been developed by software engineers as well as by artists. These applications all serve as entry points to the Web and shape the ways in which users interact. And yet, no study has so far been conducted to establish the relationships between these cross-domain developments. From the early days, the page metaphor provided a successful, initially even defining, conceptual model for presenting Web content in conventional browsers. Artists pursue diverging agendas, metaphors and spatial designs. They tend to abstain from mimicking the page and devote considerable attention to the visual, spatial and temporal configurations of their applications. Focusing specifically on these aspects, we compare five artists' browsers both with each other and with a set of recurring spatial features in conventional browsers, e.g. "page" and "tab", in order to distil implicit conceptions of the Internet, its users, and their opportunities-for-action. In these case studies (I/O/D's *Web Stalker*, JODI's *.com.mx* from their *%WRONG Browser* series, Craighead & Thompson's *e-poltergeist*, Hernando Barragán & Andrés Burbano's *Hiperlook*, and Jasmine Guffond's *Listening Back*), we analyse and address the following questions: a) How are the web browsers designed spatially/visually/temporally? b) What actions result from this design? c) What user conceptions are deducible from the possible actions? d) What Internet conceptions follow from both? Finally, we include findings from browser usage studies to introduce further parameters that help to highlight the specificities of how Internet use and access were envisioned in the browsers. We thereby hope to contribute to art and visual history, interface and Internet studies.

Introduction: Historical and technical aspects of early web browsers

The 1990s unleashed a wealth of software experiments centred around the interfaces for accessing the World Wide Web. These early develop-

ments were often guided by the need for hypertext navigation tools and have now converged into a recognisable interface schema for general web browsing. In the decades since, the ways users approach the Internet have undergone dramatic changes, evolving from browsing web directories to searching for individual websites and from supporting simple search tasks to exploring multiple web pages in parallel[1]. The opportunities-for-(inter) actions and graphical interfaces of artists' browsers can be productively analysed alongside web browsers that have been historically used by the general audience of the Internet[2]. In order to identify the features and metaphors that browsers provided in the early years of the Web, we harvested research reports and reviews in computer journals as well as memorandums issued by the World Wide Web Consortium (W3C) during the 1990s and the early 2000s.

Metaphors, mental models and opportunities-for-(inter)action

The page metaphor is already implicitly present in the early specifications of the Web (co-)authored by Tim Berners-Lee at CERN (in the form of two seminal proposals) and for the WWW Consortium (in the form of RFC documents). His early conceptualisation alludes to a hypertext page (or node) that contains highlighted pieces of text referring to other texts: these references can be clicked with the mouse in order to see the text that is being referenced[3]. Nodes need not be limited to text only, allowing for non-text media where facilities already exist[4]. Hypertext pages were seen in the short term as containers for ASCII text that could be transmitted over the distributed network and displayed on simple terminals, while the addition of graphics would be an optional extra found under the "Bells

[1] J.C. Chang, and others, *When the Tab Comes Due: Challenges in the Cost Structure of Browser Tab Usage*, "CHI Conference on Human Factors in Computing Systems", May, 2021, https://doi.org/10.1145/3411764.3445585, [accessed 12 July 2021], p. 3.
[2] Such cross-domain analyses have very rarely been conducted in the past. Cf. as an exception: C. Post, , P. Golden, and R. Shaw, *Never the Same Stream: netomat, XLink, and Metaphors of Web Documents*, in "DocEng '18 Proceedings of the ACM Symposium on Document Engineering", August 2018, https://doi.org/10.1145/3209280.3209530 [accessed 5 August 2021].
[3] T. Berners-Lee, and R. Cailliau, *WorldWideWeb: Proposal for a HyperText Project*, CERN, 12 November 1990, https://www.w3.org/Proposal.html, [accessed 7 June 2021].
[4] *Ibid.*

and Whistles" section of the original Web proposal[5]. The emphasis in the early proposals was on organising information rather than publishing carefully designed content; and on the navigation interface (i.e. the browser) that facilitates navigation while preventing the user from becoming "lost in hyperspace"[6]. The proposals resulted in the first web browser prototype called *WorldWideWeb*, which implemented the outlined principles (Fig. 1). The proposed mechanism of navigation by proceeding from node to node reveals the main opportunities-for-action of the hypertext page implemented before the commercialisation of the Web.

If we follow Berners-Lee's thinking further and across some of the RFC documents he was involved with, it becomes tangible how the abstract notion of the hypertext page was weighed down by real world compatibility requirements and turned into the page metaphor. The most telling is the RFC1866 document that specifies Hypertext Markup Language 2.0[7]. This specification "roughly corresponds" with the capabilities of HTML in common use between 1990 and mid-1994[8]. By this specification, any HTML user agent (including browsers, email clients, and so forth) that aims to conform to RFC1866 would have to consistently parse HTML code into content and layout, removing comments and whitespace (and the HTML code itself), and allow the user to traverse the hyperlinks[9]. By ridding the hypertext of its full HTML code in the process of parsing and display, a W3C-compliant browser turns an HTML document into the likeness of a printed page: separating metadata from the page content, displaying a consistent graphical layout, and providing a simple mechanism to navigate between the pages.

The *web pages* "were designed as a means for publishing primarily written text"[10] and consequently the *Web* was initially "developed for simple document delivery"[11]. Thus the task that developers might have felt they

[5] T. Berners-Lee, *Information Management: A Proposal*, CERN, May, 1990, https://cds.cern.ch/record/369245/files/dd-89-001.pdf [accessed 14 June 2021].
[6] Ivi, p. 14.
[7] T. Berners-Lee, and D. Connolly, *Hypertext Markup Language – 2.0*, November, 1995, https://www.rfc-editor.org/rfc/pdfrfc/rfc1866.txt.pdf [accessed 14 June 2021].
[8] Ivi, 3.
[9] Ivi, 5.
[10] S. Skjulstad, A. Morrison, *Movement in the interface*, in "Computers and Composition", no. 22, 2005, pp. 413-433, p. 416.
[11] P. Snyder, L. Ansari, C. Taylor, C. Kanich, *Browser Feature Usage on the Modern Web*, in "Proceedings of Internet Measurement Conference", November 2016, pp. 97-110, https://doi.org/10.1145/2987443.2987466 [last accessed 12 July 2021], p. 97.

were tackling was finding a digital equivalent to a mostly text-based document. The technical term describing this type of solution to a challenge is skeuomorphism. It describes the mechanism of incorporating familiar moves or symbolised things into new contexts or technologies in order to exploit people's knowledge of what to do with such objects[12]. Skeuomorphic elements in a user interface produce and embody functionalities, leading to opportunities-for-(inter)action.

What we mean by that has a kinship with what the ecological psychologist James J. Gibson famously termed "affordances"[13]. The concept of "affordance" is not an object's property. Instead, it is a relationship between object-related or environmental characteristics on the one hand and capabilities or operative imaginaries of an agent on the other. In other words, it is the combination of an offer ("is-for-ness") and an ability ("can-do-ness"). With some good reasons, the cognitive scientist Donald D. Norman declares the term 'affordance' applicable exclusively to physical objects[14]. He is sceptical about the possibility of applying the concept of "affordance" to computer-based representations of objects. Even the term "perceived" or "digital" affordance would not solve the issue because objects on the screen are programmable and can thus be assigned any consequence by programming. It would then become a matter of arbitrariness or convention, but would not be a property of the world. His proposed alternative concepts "convention" and "constraint" are not fully satisfactory either, however we will attempt to capture what is meant by using them alongside the terms "function" and "functionality".

What Norman indeed does emphasise, is the importance and malleability of conceptual or mental models. People form such models through their interaction, experience and understanding of the environment and they then become instances guiding future behaviour.[15] The intersubjective variants of such mental models (or "user illusions"[16] – as Alan Kay calls

[12] D. A. Norma, *The Design of Everyday Things*, Basic Books, New York, 2013, p. 159.
[13] Cf. J.J. Gibson., *The Senses Considered as Perceptual Systems*, Allen and Unwin, London, 1966; cf. [Anon.], *What is Skeuomorphism?*, "Interaction Design Foundation", (n.d.). https://www.interaction-design.org/literature/topics/skeuomorphism [accessed 30 July 2021].
[14] D.A. Norman, *Affordances, Conventions, and Design*, in "Interactions", May-June 1999, pp. 38-42.
[15] Cf. D.A. Norman, S.W. Draper, *User Centred System Design*, Lawrence Erlbaum, Hillsdale, 1986, esp. p. 46.
[16] A. Kay, *Computer Software*, in "Scientific American", September 1984, pp. 53-59.

them) can be condensed into the design of new media, in order to facilitate "going from familiar concepts to unknown ones"[17]. This is precisely what has happened with the page metaphor. Unlike a circular belt or a fanfold paper, on a formal and notional level, the page signals a conceivable end. It relentlessly secludes one finite page and its content from another, regardless of their possible semantic connectedness.

A recurring narrative touches upon a set of related metaphors that were introduced into the computer domain with the first machines furnished with a graphical user interface: desktop – document – page. The concept of the document, but even more so the mental model of the page, migrated into browser conceptions up to the point where the browser seems to become almost synonymous with a page viewer. In a seminal conference workshop in 1998, the participants decided: "Our rule is: 'When in a browser, behave like a page'"[18]. In this workshop report, much of the thinking circles around the scope of the page metaphor. Regarding the user activity (and concept of a browser) they state the following: "Browsing is when a user clicks on a link to go from one page of static information on the Web to another. This is what browsers were designed for"[19]. As long as the web browser is only concerned with HTML pages, everything works smoothly. However, as soon as it is used to serve other purposes (e.g. "application navigation") or it is nested with another metaphor (e.g. an "application metaphor"), the metaphorical cargo was regarded as confusing. Ten years later, web browsers are "still designed primarily for rendering basic pages". However, with website designs driving the integration of new features or applications in browsers, a solution is envisioned on the side of the browser,[20] not as an imperative for the web page (designers).

Despite the above mentioned highly contested usability debates and Tim O'Reilly's visionary proclamation in 2005 that the Web 2.0 was

[17] T. Catarci, M.F. Costabile, M.Matera, *Visual metaphors for interacting with databases*, "SIG-CHI Bulletin", vol. 27, no. 2 (April 1995), pp. 15-17, https://doi.org/10.1145/202511.202514, p. 15.
[18] C. Fellenz, J. Parkkinen, H. Shubin, *Resolving conflicts between the desktop and the Web: a CHI 98 Workshop*, "SIG-CHI Bulletin", vol. 31, no. 1, January 1999, 26-28, https://doi.org/10.1145/329671.329684, p. 27.
[19] Ivi, p. 26.
[20] C. Reis S.D. Gribble, *Isolating web programs in modern browser architectures*, "EuroSys '08: Proceedings of the 4th ACM European conference on Computer systems", April 2009, pp. 219-232. https://dl.acm.org/doi/10.1145/1519065.1519090, p. 219.

"going beyond the page metaphor of the Web 1.0 to deliver rich user experiences",[21] the big players in the web browser business still take the page as a dominant metaphor to this day. From an early stage, this was the subject of controversial discussions and many alternatives were suggested: *SysCo Weblet* (1995) used a "screen-based metaphor",[22] *WebStage* (1997) offered a television metaphor,[23] *The Circular Page* (2004)[24] was built upon the logic of a turning-wheel, and *Webster* (2014)[25] focused on the tagging system. When introducing their proposals, they all felt the need to explain how they deviate from the norm, that (still) is: the page metaphor. In these explanatory statements, a variety of implicit ideas linked to the "page" come to the fore: static text with hyperlinks, wordiness, heavy reading duty, linearity, layered logic of hierarchies, and a limited range of potential interaction patterns. Further notable features were the lack of visual simultaneity, movement, dynamics, flexibility and place (understood as social meeting point) etc.

Before we turn to the artistic propositions, we want to take a closer look at the central aspects of the browser interface and its functionality. This will enable us to "locate" where the page metaphor was embedded as a formal browser element. At the same time, these elements of the interface will provide a useful foil against which to set the artistic browsers.

Elements of the browser interface

Conventional graphical Web browsers – past and present – exhibit considerable similarities in their visual layout. A user study from 2010, for instance, confirms what still holds today:

> "A user familiar with the browsers of the mid-1990s would likely have no trouble using the latest versions of today's most popular browsers [...]. The

[21] T. O'Reilly, *Web 2.0: Compact Definition?*, "O'Reilly Radar", 1 October 2005, http://radar.oreilly.com/2005/10/web-20-compact-definition.html [accessed 30 July 2021].
[22] J. R. Clarke, *WWW Page Metaphor Considered Harmful*, "HCI — A light into the future: Proceedings of OZCHI 95, 4th Australian conference on computer-human interaction", 1995, pp. 264-267, p. 264.
[23] T. Yamaguchi, I. Hosomi, T. Miyashita, *WebStage: an active media enhanced World Wide Web browser*, in "CHI '97: Proceedings of the ACM SIGCHI Conference on Human factors in computing systems", March, 1997, pp. 391-398 <https://doi.org/10.1145/258549.258811>.
[24] M. Tuvich, *The Circular Page: Designing a Theatre of Choice*, in "Interactive Dramaturgies", 2004, pp. 221-229, doi:10.1007/978-3-642-18663-9_19.
[25] S. Lederhaas, K.-H. Weidmann, *Webster: A New Information System for the Web*, in C. Stephanidis (ed.), *Proceedings of HCII 2014: Posters, Part I*, Springer, New York, 2014, pp. 180-185.

main navigation mechanisms – hyperlinks, the back and forward buttons, the URL bar, bookmarks, and the history – have remained almost unchanged"[26].

That is to say, the browsers' graphical user interface designs have remained surprisingly stable and consistent despite the dramatic changes in the underlying technology and the Web itself. While their functional layering has increased over time, there are still only a few elements that determine their nested structure in terms of spatial layering: the frame within the browser window, the display field (viewport) within the frame, the page within the display field, and – where applicable – the tabs that hold a conceptual as well as spatial middle ground between the first two elements. Let us briefly zoom in on these elements.

Frame. The frame contains a) icons embodying functionalities, b) the URL bar and possibly c) the scroll bars, inviting users to click, type or drag. Except for the scroll bar that accommodates the size of a web page, the frame is the only element that remains unaffected by the web page displayed. The frame is well-named because it surrounds a distinct display field reserved for displaying the content of the websites. Since the earliest web browsers, users have been able to scroll vertically (and since 1994 also horizontally) through the HTML document rendered in this field. The scroll bar is used to navigate *within* the called website; the 'reload' button allows users to jump to the beginning of the (reloaded) website[27]; the rest of the frame functions concerning navigation allow cross-website orientation and switching *between* websites.

Display field. The display field or viewport is an area where the web page content is shown. Its size depends on the size of the screen. It is surrounded by the frame and can show a variety of data forms. When looking at one of the earliest web browsers, *ViolaWWW*, released in 1992, the icons on the top right clearly indicate what was supposed to appear in the display field: pages emphasised as such by the dog-ear (Fig. 2). The display field and the page that is supposed to appear therein, are easily discernible as soon as a scroll bar is present as it indicates a difference in size.

Page. The page presents itself as a flat rectangle, more often taller than it is wide, and by default light grey or white (in Fig. 2 light yellow). To many,

[26] P. Dubroy, R. Balakrishnan, *A study of tabbed browsing among Mozilla Firefox users*, in "Proceedings of the SIGCHI Conference on Human Factors in Computing Systems", April 2010, pp. 673-682 https://doi.org/10.1145/1753326.1753426, p. 133.
[27] Ivi, p. 681.

this mental model feels so natural they lose sight of the fact that it is actually a design decision. This oversight also seems to become evident in the way that participants in user studies tend to refer to website contents or tasks rather than the interface itself.

Tabs. The browser tabs are designed to be part of the frame but occupy an interesting position between content, frame and usage. The tab feature can be found in browsers from as early as 1994, when BookLink Technologies Inc. implemented it underneath the horizontal scroll bar at the bottom of the browser window. A contemporaneous magazine article promoted it like this: "InternetWorks' multithreading capabilities let users open many separate connections within one window. For example, users can conduct several searches and downloads on the Internet"[28]. The dominant browsers established the tab feature in the mid 2000s. Tabbed browsing allows users to navigate the Web while acting as a temporal history and memory as well as creating perceived possible connections between content actually scattered over different pages.

On the visual level this design element can be almost invisible if just one tab is open, or it can become an ornamental strip. In practice, a row of tabs conserves a lot of information about its user's activities. As an invention, tabs met the user need for an efficient way to revisit websites they had previously found (accounting for 58% of user activity in the mid-1990s)[29]. They generally provide non-linear access to the user's browsing past and may lead to different or even contrastive habits. Some users may find them helpful in managing and keeping track of their own work progress, providing a space for "the big picture of where everything is at"[30]. Others use declutter tools to stack tabs as the latter add "another level of complexity to the web browser interface"[31]. resulting in a need for users to manage their tabbed browsing.

Although studies show that "[u]sers do not spend a great deal of time interacting with the GUI widgets of their browsers relative to the amount of time they spend engaged in things like reading, visual search, and

[28] K. Rodriguez, *BookLink browser to become freeware. Move follows Mosaic's lead*, in "InfoWorld: The Voice of Personal Computing in the Enterprise", 7 November 1994, p. 56.
[29] L. Tauscher, S. Greenberg, *How people revisit web pages: empirical findings and implications for the design of history systems*, "International Journal of Human-Computer Studies", no. 47, 1997, 97-137.
[30] P. Dubroy and R. Balakrishnan, *op. cit.*, p. 681.
[31] Ivi, p. 67.

waiting"[32], the features described above can be shown to have a significant bearing on the browsers' offers-for-action and therefore directly influence the scope of user behaviour.

Five artistic web browsers

Standard browsers are expected to fulfil a number of basic functions, namely: retrieval of the requested website's assets, negotiation of file formats and rendering of the layout consistent with the WWW Consortium specifications. This requirement can be compromised by certain requested features (e.g. Opera's website compression[33]; reconfiguration as print friendly version), policies (limiting access), temporal incongruencies (when new standards are not yet supported) or tactics with proprietary software (one line of combat between rival browsers). However, in general, conventional browsers are supposed to provide the user with access to the website's form and contents as envisioned by the website designer. With artistic browsers, this is not to be expected. Artists are neither much concerned with how web page designers wish their creations to be displayed, nor are they particularly excited to follow guiding principles to ease and please, e.g. user-friendliness.

When looking at artistic examples, we found the defining principle that makes these applications identifiable as web browsers is not the page metaphor, but the visual-operative presence of a URL bar. Most of the other functional fields common in conventional browsers are optional. As compared with conventional browsers, the relationship between the time users spend engaged with the browser interface versus the web content retrieved using it shifts in the artistic domain. What then are users busying themselves with and what are the implications and consequences? Could there be other agendas for web browsers that go beyond serving as an unobtrusive, supportive see-through medium?

To address these questions, we analyse a selection of art browsers fo-

[32] M.D. Byrne and others, *The tangled Web We Wove: A Taskonomy of WWW Use*, "CHI, 1999, 544-551", https://dl.acm.org/doi/pdf/10.1145/302979.303154, p. 550.
[33] S. Orgera, *How to Turn on Opera Turbo Mode*, in "Lifewire", 17 February 2021, https://www.lifewire.com/activate-turbo-mode-in-opera-for-linux-mac-and-windows-4103691 [accessed 31 July 2021]; R. Voigts, S. Christmann, S. Hagenhoff, *Mobile Web Browsers*, University of Göttingen, 2011, http://webdoc.sub.gwdg.de/ebook/serien/lm/arbeitsberichte_anwebus/2011_01.pdf, [accessed 2 August 2021].

cusing on their spatio-visual design, which results from and has consequences for further important dimensions, such as information density and quality, functionality, and temporality. Our hypothesis is that the specific interplay of all these dimensions and facets in each case paves the way for special opportunities-for-actions. Therefore, we are going to unpack this constellation in each of our five case studies.

Except for the more recent *Listening Back* (2019) by Jasmine Guffond, all these browsers were developed in the first net art era around the turn of the millennium and therefore contemporaneous to the observations documented in the early sources we discussed above: I/O/D's *Web Stalker* (1997-1998), JODI's *.com.mx* from their *%WRONG Browser* series (2000), Alison Craighead & Jon Thompson's *e-poltergeist* (2001) and Hernando Barragán & Andrés Burbano's *Hiperlook 1.0* (2002). Our case studies focus on the following four questions: a) How are they designed spatially/visually/temporally? b) What actions yield from this design? c) What user conceptions are deducible from the action possibilities? d) What Internet concepts follow from both?

The Web Stalker

One way to perceive the Internet is as a system that exhibits a certain kind of depth. This notion opens up a number of approaches. One possibility would be to render the connected structure into something like the *Map* function of *Web Stalker* (Fig. 3). This browser's visual output brings to life – whether intentionally or not – Tim Berners-Lee's unrealised early vision of how a web browser could be used to aid the analysis of projects and organisations. In a seminal proposal, he anticipated that databases, e.g. at CERN, would grow so large and interwoven that they would be very difficult to perceive in their entirety[34]. Berners-Lee proposed making a three-dimensional model with people represented by spheres and the connections between people by strings. Users would interact with that model by "picking [it] up" and "shaking it, until you make some sense of the tangle", thus revealing "the real structure" of the organisation[35]. This description seems to align with the *Web Stalker*'s *Map*.

[34] T. Berners-Lee, *op. cit.*, pp. 11-12.*op. cit.*
[35] *Ibid.*

After opening *Web Stalker*, the user is confronted with an empty black application window. There are no indications of possible interactions. The user must consult the help.txt file to find instructions on how to use the program[36]. To access the functionality of the application, the user must create windows[37] inside the main application window by holding down the left mouse button and dragging to define the desired window size. This forms another blank rectangle. Pressing the right mouse button inside this field opens a pop-up menu that offers the user six functions to choose from and then assign to the created window. The functions selectable in the menu are called *Crawler*, *Map*, *Dismantle*, *Stash*, *HTML Stream* and *Extract*.

To gain access to any other processes and functions the browser provides, the user must first create a *Crawler* window. Within this window, they can enter a URL through the *Crawler*'s pop-up menu. The *Crawler* window shows the search, but not its results. Other types of windows are needed for the content of a web page to be shown. If the user decided not to open any other window, the crawling process would be performed but only documented by the animation of a bar that is split into three equal sections with a dot moving along them. There is also a text field providing information about the status of the programme's processes.

Maybe the most prominent function of the application is the *Map*. It creates a visualisation by mapping "the links between HTML documents. It shows references to individual HTML documents as circles and the links between them as lines", starting with the URL typed in by the user. When a circle element is selected (by clicking on it), its URL can be seen in the top left of the *Map* window. When a circle element is selected, a second circle will appear inside the first one. The *Map* can be saved within the pop-up menu of the function window, which also provides offline access to the visualisation.

The *Dismantle* function "is used to work on specific URLs within HTML documents". A circle from the *Map* can be dragged into the *Dismantle* function window and it will display all URLs referenced within the HTML

[36] Unless otherwise specified, these very detailed instructions are the source of our following quotations, cf. M. Fuller, S. Pope, C. Green, *I/O/D 4 The Web Stalker Version 1.0 for Windows 95 Help document (accompanying the executable of Web Stalker)*, 27 November 1997, https://bak.spc.org/iod/iod4.html [accessed 2 August 2021].
[37] The artists call 'windows' what we would name 'frames'. For the sake of consistency when quoting them we use their terminology.

document, also representing them as circles. "Clicking on any of these circles reveals the URL of the resource as text in the top left hand side of the *Dismantle* window."

The *Stash* function provides a flexible way to save the URLs, dragged as circles from the *Map* or *Dismantle* windows, as a document, that can be also read by conventional browsers. For that, the *Stash* has to be saved as HTML, which is an option in the pop-up menu of the window. The *Stash* function is one of the two functions that can be opened several times in the application window, to allow a separation "of resource documents for different subjects during one web session for instance".

The *HTML Stream* function shows "all of the HyperText Mark-up Language [. . .] as it is read by the program". Using the application now, it is not clear how the stream was intended to be displayed initially, as only the first element's HTML code seems to be displayed.

The *Extract* function shows all the text from a URL dragged from the *Map*, *Stash* or *Dismantle* windows as a scrollable text file. The text in the window can be saved by choosing this option in the pop-up menu of the window. Like the *Dismantle* function the *Extract* function can be opened several times.

In addition, the user can choose a background colour by an option-click in the main window. This opens a vertical colour panel offering a choice of six colours: dark grey, ultramarine, purple, dark green, dark olive and black.

As the browser window has no initial layout, the user is forced to create their own structure. This offers the maximum range of opportunities for arranging and layering as the "windows can be resized and repositioned at any time during use of the program. They can also overlap each other" even to the extent that one window can superimpose itself on another completely. The perceivable spatial depth, however, is relatively shallow. There will probably always be a negotiation between maximising clarity on the one hand and the amount of information displayed on the other. If all the functions are open, potentially even with more than one incarnation of some of the functions, the display can quickly become crowded. Windows have to be resized, relocated and their layout changed to be able to see everything[38]. As Joseph C. Chang and colleagues state in their user study in 2021[39], allowing users to arrange their documents freely in

[38] Cf. P. Dubroy, R.Balakrishnan, *op. cit.*
[39] J.C. Chang and others, *op. cit.*

a 2D or 3D space can even be detrimental, especially if they are not already familiar with the space or "when their mental models evolve rapidly as they consume new information"[40]. From these findings, it can be concluded that users will spend more time reconfiguring the display to make the desired information accessible than they would without the freedom to choose an arrangement of their own. That especially is to be expected when they are first confronted with *Web Stalker.*

In terms of the number of display elements, *Web Stalker's* windows have some similarities with tabs in that it is possible to create a stack of windows. Unlike tabs, however, the spatial configuration results in an overlap and thus a reduction in the level of overview that multiple windows would provide. In contrast to multiple opened windows or tabs, the different windows not only allow different web pages to be displayed, but also provide a variety of functional lenses for the same dataset.

In a study from 1995, the most commonly cited reason for working with tabs was to have a "cleaner, more organized, or less cluttered" situation than when using multiple windows. Some participants "said that it was helpful that tabs kept their web browsing 'all in one place,' whereas multiple browser windows would be interspersed with other application windows"[41]. It is interesting that users prefer to collect all the information they are seeking within the confines of one application window, a concept that is developed within the *Web Stalker* application. The spatial configuration of its stacked windows differs from that of tabs. Although created by the user, the layout and location of the latter is pre-determined, set from left to right in an orderly horizontal row, "one next to the next".

InternetWorks, the first web browser using tabs in 1994, "uses a tabular or folder metaphor for resources scattered across the Internet. You can navigate either via a single large screen [...] or by selecting from predefined and user-created folders and index card-style forms. [...] InternetWorks also has a decent system for connecting to multiple sites at once – a command lets users split their screen into any number of windows. Unfortunately, resizing these panes can only be done horizontally or vertically, depending on how you originally created them"[42]. In this respect the *InternetWorks*

[40] Ivi, p. 2.
[41] P. Dubroy and R. Balakrishnan, p. 677.
[42] A. Gaffin, *Take a walk on the Web site*, in "NetworkWorld: The Newsweekly of Enterprise Network Computing", 8 May 1995, pp. 57-59, p. 59.

browser can be seen as the mental 'transitional fossil' between tabbed browsing and *Web Stalker*, even though it predates it.

The balancing of information and overview seems to be the central opportunity-for-(inter)action and creates a different structure of display, information and depth for each user. The visualisation of the *Map* itself has similarities with the spatial layering described for the interface design. The circles depicting the single pages are displayed in an overlapping way, although the user is not able to rearrange the emerging structure. Depending on the number of HTML documents connected to the user's URL input, the drawing can be very dense and hide circles behind a multitude of added circles. But there also is a layering of information. Users can customise the direction and depth of the information displayed by selecting the windows they open and even opening multiple windows of the same function and arranging them at will. This provides different views of the accessed web page and the Web itself. On a more abstract level, there is a layer structure in this network tree representation that starts from the central (chosen) node with generation after generation reaching out into the next levels of its periphery. On the one hand, this gives the impression of providing an exclusive behind-the-scenes view of how the non-linear net is actually organised. On the other hand, it only shows the section of the net close to the user's trail. This map is blind to everything that is not directly linked to the starting node. Constructing the Internet this way renders it as a graspable and ostensibly finite space. Geometrically, circles have the visual potential to be a "container". This containing capacity (for links) is not shown directly, only as further lines departing from its margin. The *Map*'s mapping of the abstract link structure into a spatial representation theoretically could result in a navigable world, but it turns out that this two-dimensional image with almost no depth offers very few possibilities for interaction. As described, users can select elements and drag them into other – functionally specified – windows to be analysed. This is similar to the scientific process of extracting samples; here, these are chosen and collected under the complete control of the user, with no further intervention by the application itself. Sampling is used in a variety of research fields and refers to the isolation of small components of a potentially complex system. The resulting reduction in complexity automatically leads to a reduction of informational depth but a gain in terms of clarity and focus. Thus, it can be argued that in the visual representation of the Web provided by this browser, the

same balance between information and overview is negotiated as with the interface's opportunity-for-(inter)action.

%WRONG Browser: .com.mx

%WRONG Browser is a series of browser applications by the artists' collective JODI (Joan Heemskerk and Dirk Paesmans). JODI developed eleven browsers for the series, each named after the top level domains to which their automated web access is restricted. Each browser creates a different and visually very specific composition using an artistically created colour scheme, graphical elements, the display of the URLs and the unparsed HTML code of the accessed web pages in a constantly animated collage. The domain ending already limits the range of web pages that can be automatically accessed. The two or three letters that are randomly inserted to complete the URL restrict the parts of the Web that can be accessed even further. Although users can manually type in any URL they wish, this automated restriction is the dominant creator of the visual setting for the work. Due to the automated URL selection, the Internet is already fragmented into a few slices that are the accessed pages. The Internet here feels like a randomised sampling demonstration, offering the user a disconnected set of different tasters. The Internet is accessed and thus represented but it is visually absent (as a network). A remnant of the network, as an immanent part of the program's perceivable elements, is the sonification and visualisation of the connection process that describes the retrieval of the web page data. Thus, the user can gain an abstract idea of the depth to which the browser reaches out but cannot grasp its dimensions within the work.

As we have seen, even though *Web Stalker*'s interface is customisable, there is still a distinct division between content displayed and the framing structure that surrounds it. In contrast, *%WRONG Browser* offers a shared space for both the visual and informational aspects. The application opens to full screen. There are no framing elements to signal any separation. There is no functionality discernible from the displayed elements and over a few seconds the browser will start its automated processes. The only recurring element providing a hint of a functionality is the URL field that promises the user an opportunity to enter text. But as the URL fields in all *%WRONG Browsers* are either moving constantly or at random intervals, in

some cases very quickly, users would be unlikely to associate this functionality with these fields immediately.

.com.mx. We are going to take a closer look at the *.com.mx* variant of the *%WRONG Browsers* (Fig. 4). In terms of both its programming and visual structure, this browser is complex enough to show the prominent aspects of the series. However, its internal structure i.a. documented in the code, is still comprehensible enough to include its relevant parts here (Fig. 5). Like most of the browsers in this series, it has five compounds. In this case, a compound comprises a button, a green vector graphics shape, a URL field, an HTML field, and an HTML-style display field. The first four of these elements are attached to each other, the last one is partnered with the cursor. The five compounds can display up to five web pages at the same time in the same browser window. Each element has specific properties that partly determine its behaviour as well as the ways in which users can interact with it. The button is a white horizontal bar that moves automatically. It leaves a visual trace by being newly drawn at each position without being erased at the previously occupied position. The trace has no operativity while the actual button has. It is possible to drag the active button with the mouse and to double click it. This results in the emergence of a new URL in its connected URL field. The URL field is spatially connected to the button as well, so that it moves with it when the button is moved. It is also editable insofar as it is possible to write into the field. Initially, a string of random two to three letters is inserted with the constant domain ending .com.mx. These random URLs can therefore be changed by the user and accessed by pressing "enter". The only other key that affects the field is the space key that results in aborting the input. This field is fixed to a width of 21 characters. It does not limit the user's input to the format of a URL or a specific number of characters and can therefore vary in its height and seems to be infinitely expanding out of the accessible screen. The text can be selected and deleted. Selecting a text turns the selected background turquoise and the text black, inverting both the red background of the field and the white text.

The HTML fields show the HTML code of the accessed page. They move together with the button and the URL field. They cannot be changed by the user other than by moving them with the button. If the HTML code has a specific first line, a previously empty HTML-style display field shows a version of the HTML code that is not changed in font – as is the case in the HTML field – and even contains rendered layout elements as it uses

the hypertext styles. In four cases, this HTML-style display field can get spatially connected to the mouse and therefore gets dragged along while the user moves the mouse, which in one case is fixed at the left edge of the application window. Like the button, the HTML-style display fields leave a trace and therefore can be used to 'paint' the browser window. The user has no control over the connection or disconnection of this field to the mouse cursor as they cannot provoke or end the connection by any means. The URL field and the HTML field attached to it have transparent properties but a red background and, if they overlap, will draw on top of other elements' traces either with the text or the field background. The green line connects two buttons. While the buttons are moving due to the programme's automation or user interference, they will keep this spatial connection to the green lines which in turn leave traces of their own. Together with the two other types of elements leaving traces, this creates a composition that can be produced completely autonomously based on the randomisations used by the programme. It can also be a co-creation with the user. In any case, the composition will never be settled, as the application's automated processes are constantly changing and adding to the design.

Two aspects of these drawing activities seem to be of special interest.

First, even though the user can determine the movement of the button and all other fields spatially connected to it, one element is beyond the user's control. Every button (or at least four of them) will move automatically at some point during a runtime but can be specifically selected to 'paint' the trace of its elements. In this case, the additional HTML-style display field gets attached to the mouse cursor by technical chance and will leave a trace as soon as the user moves the mouse. At that point, therefore, the only choice that still resides with the user is not *whether* but *where* to actively leave a trace.

Second, the button – as the crucial element for allowing the user to engage in the creation of the emerging composition – does not exhibit its functionality at all. On the contrary, it rather seems to conceal it. It is a button by definition, in the source code and in its functionality. But in design it has nothing in common with a button in conventional applications, and certainly not with those at the time of its development. The background of the browser window is red, the trace left by the buttons is white – a colour more commonly associated with a background colour. Observing the browser's behaviour, therefore, might create the impression that the white

bar is a blank space rather than an active element with which to interact. Consequently, it might take some time before the user discovers the interactive potential of this element – if indeed they ever do. The interaction might remain on an observational level during the whole browser session.

The more obvious points of interaction provided by the browser are therefore the URL input fields for accessing specific websites. This access is only possible, however, until the timed automation creates a new random URL to be accessed, thereby overwriting the input of the user. While the possibility to type a URL could be seen as the minimum requirement for the application to be perceived as a browser and for the user to have any kind of self-governed access to the web, the above mentioned interaction with the white button is rather specific. Where conventional browsers render a page as defined by the HTML statements (among other things), the *.com.mx %WRONG Browser* creates a design to which users can contribute. The potentially five displayed web pages and their content have little bearing on the composition. The visually and compositionally dominant elements are the ones that leave traces and they, more than any single web page, add to the creation of the dynamic design that is always the combination and connection of the five compounds.

There is no conventional interface and there are no visual clues to indicate that an interaction is necessary or even possible. But once a user has ventured into experimenting on what they can do, the automated processes continue to make it difficult to determine what aspects actually can be influenced or interacted with. Here, almost all the navigation options afforded by conventional browsers are absent. Links are inactive, there is no "back" or "home" button and typing in a specific URL can be difficult as the automation will take over at random moments and overwrite the user's input. As *.com.mx* radically reduces the breadth and depth of the functionality central to conventional browsers – search functionality is minimalistic and the opportunity for receiving the retrieved web content is either not displayed, or partially in its raw HTML code format – users may settle for alternative attitudes. They may adopt a contemplative approach and observe the events on the screen. As there is always something unforeseeable going on, the user can sit back and observe the autonomous behaviour of the program. The observability of this spectacle is certainly a strong feature of the application. The reduction might even pique an exploratory behaviour in some users, for instance comparing the five compounds and looking for behavioural patterns. While this is not necessarily

goal oriented, it focuses on the application itself, a conventional browser might not yield such behaviours. In the exploration of the Web under circumstances where the browser heavily restricts the path to information, this may become a personal confrontation with the browser itself. The user may come to the conclusion that they need to fight to get their say. There is no way to claim control over this machine-centred browser as it involves a great deal of xeno-techno-cratic ruling. For the user immersed in *.com.mx*, too much happens in parallel and too fast for the human mind, rendering this human/computer-interaction stressful. The narcissistic insult becomes complete, when – after a long struggle – the user discovers that there is indeed a user-friendly feature built in. One URL field never moves. It is static so the slow user can 'catch' it to insert an address. The other four are for a faster automated force.

However, loss and defeat are not the only experiences this browser has to offer the user. Something can also be gained by reconceptualising what is at hand. In other words, the user can overcome or outsmart the constant feeling of inferiority by changing the agenda. In addition to passively observing the application's behaviour, it is also possible to co-create the composition in a manner reminiscent of Adrian Ward's *Auto-Illustrator* (2001). There are certain parallels to drawing software, as the user can select the button element and use its form as a stamp to leave traces. Moreover, the button connection to two lines of the green "pentagon" can be used to produce inclined vector lines. The new and challenging aspect of this is that every painting tool becomes coupled and is productive simultaneously if the user picks up the white button: The resulting (compound-as-)composite brush – consisting of a white-painting horizontal rectangle, two green-painting vector lines, a red-painting frame with unaffected black text on top, and a mainly black- and blue-painting text directly attached to the cursor – effectuates a) the predominance of a certain colour in a certain region of the screen and b) a difference in the outcome depending on the direction of the paint stroke (even if the digital colours are completely flat).

There is an immanent layering of the compounds and elements of the browser's display that creates visual structures that have just a temporal depth (see Fig. 5). Due to their transparent properties, the moment an active element moves above other elements the layering becomes visible through movement. Anything that stays still gets overpainted. The user needs to be familiar with the expressive scope of each painting tool to un-

derstand the superpositioning that occurs. The outcome as such does not emphasise spatial depth. However, this can be deduced from the genesis. As the composition is in constant motion, with or without the influence of the user, the immanent impression of a layering process is strong. Thus, as a performing painting programme, JODI's *.com.mx %WRONG Browser* offers an opportunity for experimentation. Besides the complex brush, a second innovative aspect when using the browser as a paint programme, is the fact the user-painter may continue engaging with the programme while actually waiting for serendipity. This may be waiting for the co-creation to produce an interesting composition – which would have to be captured by a screenshot immediately – or waiting for an interesting "paintload". As described, web content becomes attached to the cursor and can then – in an interesting text-to-image conversion – be painted with via the movement of the mouse. Normally, this content is text only – as the browser omits all images, videos or sounds – resulting in much black (for normal text) and a bit of blue (for the links, see Fig. 5). Sometimes however, a website's HTML code is rendered not only as letters but also as graphical layout details which enhance the form and colour palette in unexpected ways.

Hiperlook 1.0

Hiperlook 1.0 is a web browser created by the artists Hernando Barragán and Andrés Burbano in 2002 (Fig. 6). After opening *Hiperlook*, the user sees a frameless, discreet, regular grid drawn with thin white vertical and horizontal lines on a black background. In addition, an orange coordinate cross moves with the tip of the cursor. At first sight, flatness prevails. Moving the mouse out of the centre moves the grid. The closer the mouse comes to the border of the screen, the faster it moves. On the bottom left, the user can enter a URL. Pressing "enter" then shows the textual and pictorial content of the corresponding website within a display field that is defined by four small white squares at its edges. The website contents are stacked on top of each other and pinned together on the left upper corner of this field. They are suspended in an only modestly defined space. Based on the time an element needs to traverse this space and reappear again, we determined that horizontally about two thirds and vertically only about 45% of the space is visible on the screen. In other words, the view into this space is partial and we are dealing with a non-oriented topology (Fig. 7).

Furthermore, whether this space feels huge or small depends also on the loaded web content, i.e. on the picture's proportions.

Whereas the displayed web page's HTML text background is transparent, the stacked images are not. It turns out that the slower the Internet connection, the more completely the user gets to see the web page content: as it loads one image after the other in the order of their appearance, these then overlap partly or completely depending on their proportions. In a way, the amount of data is temporalised. The more images are loaded, the longer this building process can be observed. However, it normally lasts only a few seconds at most, not enough time to appreciate what this unorthodox "slide show" is actually presenting. The HTML source code text overlays the image stack and can be scrolled if it is longer than the predefined display field. The web page seems to be shown as a kind of "boxed" source material. In the context of the whole screen, this bundle is attached to a separate flat coordinate filament parallel to the computer screen. Each website retrieved is given its own transparent sheet in this space. These are staggered in an orderly display system (unlike Marcel Duchamp's exhibition design that also combines lines and exhibits: *Sixteen Miles of String*, 1942). In *Hiperlook*, it is more like a collage of loose windows but spatially arranged. Navigation and focusing on a specific bundle are performed by hovering the cursor (not by clicking as there is no functionality attached to clicking). The focus switches from one web page bundle to the next as soon as the cursor is hovered over the upper part of a collaged bundle that is not yet at the front, resulting in it swapping layers with the page that was previously at the front. This disturbs the spatially indicated order of appearance of the web pages and creates a more complex construction of the space as it enables the user to weave the sheets. As hovering is the only form of navigation for all dimensions of the space (global system behaviour, focusing on a web page bundle and scrolling the HTML text of any specific web page), moving the cursor away from the centre of the screen will always influence the configuration of all the elements in the global system. The only exception is when an HTML display field reaches the centre of the screen simultaneously with the cursor, while the cursor needs to be slightly decentred with respect to that field. Only then, and only for the time it takes to scroll through the text, will there be no effect on any other element than this one.

Changing the viewpoint (operationalised as cursor position) allows the user to rearrange the whole, but more flexibly than in a gallery. All layers

are dynamically linked to the cursor, leading to a meditative floating effect – a fly-through. However, creating specific collages of the web page bundles (e.g. stacking all of them on top of each other; or eliminating the black areas by covering the whole screen or viewport with images etc.) requires very targeted navigation and that turns out to be challenging. Although users are given the power to be the prime movers steering everything in this world, they soon realise that the complexity of the task may exceed their capacity to calculate the delicate dependencies. As all the web page bundles travel at different speeds and degrees of transparency, it is like conducting a quirky orchestra whose members all interpret the one sole trajectory of the cursor-baton slightly differently. Reflecting on the overall sensitivity, the user may end up moving the mouse patiently, slowly and very little, thereby observing all the elements of the desired configuration and correcting slightly if anything moves away from its assigned target. However, this hard-won configuration is stable only if the user can finish assembling it with the mouse resting at the same time in the middle of the screen. For users possessing a drive to tinker, this browser offers a challenge if they are minded to invest in that mode. But sooner or later it is more likely that the user will lose patience and decide to either go for randomness by sharply accelerating and then stopping; or for meditative relaxation by choosing a pleasing velocity, and then sitting back to watch the chosen web pages float in this artificially designed cyber space.

It is worth noting that it is only due to the grid configuration that we perceive motion, acceleration and, when several web pages are open, layers of depth. *Hiperlook* encodes the browsing history in its spatial configuration. The longer the time that has elapsed since the search result was loaded, the less prominently it is displayed (its colour changes from yellow to grey). Each retrieved web page is added to the screen. When each new website is loaded, all the previous ones retreat a step further into the background. At the same time, they are less affected by the movement of the mouse. Thus, the farther back a website, the slower it is (which contributes much to the spatial understanding) and thus paradoxically the easier to catch. Hovering over the upper four lines of displayed code of a web page bundle brings it to the foreground and frees it from all eventual overlapping entities (lines as well as web page bundles). However, this renders it highly sensitive to any mouse movement once again. Thus, the user may choose between a visual or a temporal fleeting of the web page bundles. The web pages are clearly not conceived to rest.

Despite its spatial properties the geometrical structure does not seem to try to capture the networked character of the Internet itself. Rather, it seems to propose a certain method for displaying information. There is no connection between the grid structure or its layers that is directly, indexically connected to any outside information or data. It is an offer to organise information in a way that might also hint at a concept of the Internet. But this concept would be fully integrated into the browser interface. The lack of connection to any point of reference external to the work or any connections between the elements loaded into the infinite box keeps this space neutral or even isolated. A virtual space is created in which the data is displayed and can be navigated. The navigation of this closed-off environment lures the user into weaving through it. The experience seems focused on the perception of traversing space between the elements. The elements themselves stay flat though and therefore highlight the abstract nature of the space. It is like cutting atomic parts out of the Web and putting them in an artificial environment to observe them and walk amongst them.

e-poltergeist

e-poltergeist (2001/2012[43]) is a web-based work by Alison Craighead and Jon Thomson (Fig. 8), that is still to be found on the page http://www.thomson-craighead.net/po1t33_ge15t/ even though with the termination of *Adobe Flash* much of its functionality is no longer accessible. Here, by clicking the link to an English or French version, the HTML codes of the linked site would create a potentially never-ending series of events, mainly comprising automatically opened new tabs and windows with the latter spread across the user's desktop. Several tabs open showing the live search results of pre-programmed searches on *yahoo.com.uk* that all seem to be composed by a lonely person asking for help with sentences such as "Can you hear me" or "Please listen to me". With an almost constant score of sounds and noises, windows open at seemingly random places on the user's desktop, flicker and switch places. A steady succession of new tabs is also created. The behaviour executed by the browser is called forced

[43] In its intended functionality, the user would visit the website of the 2012 reworked version of the original version which was commissioned for the exhibition 010101 held at San Francisco Museum of Modern Art (SFMOMA) in 2001.

branching, which refers to the browser allowing web pages to open tabs and windows on their own, thereby taking away the user's control over the accessed pages and contents[44]. *e-poltergeist* embodies an interesting scatteredness. It sends distributed messages about corporate Internet companies in the tab headings, it anchors further deep references to artists in the source code, it appropriates and thus conserves ads in the small fleeting windows, it flags the application with a favicon design. The visual layering is that of overlapping windows. Their behaviour seems erratic, especially when the user attempts to interact with the work. Therefore, no constant concept of depth is created. The visibility of elements more closely resembles activity than a location in a potential virtual space. The depth that is created is one of time, almost musical in its creation of rhythm. Besides the sounds that come seemingly from nowhere, the most eye-catching and distracting features are the small advertisements, nicely separated from the rest as they were in the good old days.

It is this erratic behaviour that provides the primary opportunities-for-(inter)action to the user when first confronted with the work. Any attempt to take control over the moving, overlapping and appearing windows, is a constant undoing of the work's central behaviour. It is like fighting the Internet in its most tedious characteristics. But by sticking with it and allowing the need to control what happens on their own computer screen slip into the background, users can certainly adopt a variety of strategies for dealing with the browser. One strategy can be to test one's own effectiveness by systematically trying to engage one instance of this work after the other. After a while, the user understands that trying to catch, keep or close these hyperactive windows is doomed to be a completely futile enterprise. The ones that were indeed closed, keep coming back and haunting the clear space. It might feel like fighting the mythological Hydra. The next trial could involve opening a new tab or using an existing one to enter a search term or URL. This is possible, although an automated next tab call soon presses forward, pushing the user's own search into the background. Keeping track of one's own tab seems like a full-time occupation, preventing every other significant action one might have in mind. It becomes a question of domination. The only lasting effect that users can

[44] Cf. J. Huang, R. W. White, *Parallel Browsing Behavior on the Web*, in "HT '10: Proceedings of the 21st ACM conference on Hypertext and hypermedia", June 2010, https://jeffhuang.com/papers/ParallelBrowsing_HT10.pdf [accessed 20 July 2021], p. 4.

achieve through their own interventions is to determine where in the tab row the next automation takes place. After that, what remains is more an explorative and interpretative behaviour: trying to read the headlines of the tabs (which becomes increasingly difficult as more tabs are opened), attempting to decipher the favicon motif, having a look at the source code, letting oneself be lured to the search results that the work's own search phrases determine. Finally, the user can go with the flow and, for instance, play all the videos that the automated *e-poltergeist* search brings up in the *Yahoo* list in order to enhance the soundscape into a cacophony that regularly sets in. Adopting this attitude would imply 'playing' rather than 'browsing' on one's own. By limiting the user's interactivity, a kind of desperate symmetrical situation is created: the user cannot really get their message through yet neither can the poltergeist (from the other side). The poltergeist poses its requests on a regular basis, however that regularity can be disturbed by the user's activity. Whenever the user gains some agency, the 'person' crying out for help is lost from sight: a devastating situation for someone who would want to reach out and provide assistance. In a slightly different way than JODI's *%Wrong Browser*, it leaves the user overwhelmed and helpless. But where *.com.mx* offers a focus point for user activity, with *e-poltergeist* the only behaviours left in the end are defensive actions to get rid of the ghost in the machine. The Web here is just a conceptual background but not a layer of display. The layering is visual as in the *%WRONG Browser*, but primarily it is temporal. The application is distributed. The single functions and calls creating the choreography of the work do not exist in one place but use the connectedness of the Web itself. It is like an automated version of the Web compressed in time and space.

Listening Back

The *Listening Back Chrome* extension works very differently in terms of its layering from the previously discussed artistic browsers. It has no visual representation except for a menu that allows the user to change certain parameters in a separate browser tab (Fig. 9). Otherwise, all visual output is kept within the specification of the *Chrome* browser, inheriting all of the layers and opportunities-for-action the conventional browser provides.

The extension also differs from the other artistic browsers analysed here

because there is no aspect of the rendered web page that is withheld or even changed in comparison to what the conventional browser displays. The extension only adds to the existing function and information layers. The application sonifies web pages' usage of web cookies. It tracks when a new cookie is inserted, deleted or overwritten on the computer and produces a sound in correlation. For a list of 62 domains and their related cookies, there are specific signature sounds. For all other domains, a semi-random frequency is generated. It also tracks the persistence of the cookies and this has a logarithmic correlation with the time the specific cookie sound is played.

The menu allows the user to determine the scales and pitch of the sounds. The volume of first- and third-party cookies can be adjusted independently, and the user can type in a list of domain names to adjust their volume and pitch separately.

The strict correlation between the sound and the specific action performed by the website regarding cookie insertion etc. makes the addition of information extremely prominent and uses sound as its medium. The extension renders an omnipresent aspect of the modern Web, with which everyone is familiar, perceivable as part of the actual user experience. Theoretically, the user does not have to change their behaviour when using the browser. However, the nature of the sounds would lead us to expect a significant change in behaviour as long as the user does not put the created sounds to the back of their consciousness.

Activating the extension results in the creation of sounds when a website is accessed because this is the point at which new cookies are inserted. As the duration of the sound is correlated to the cookie's longevity, for most web pages the moment of first access will be the one when the most sounds are played. If the user stays inactive, the score's intensity will decrease and, in most cases, at some point the sound will cease completely. Depending on the visited page, the initial overwhelming sound composition could potentially lead to just such a pause in the user's activity.

With most commonly visited websites, the interaction of the user with the content will trigger new cookies to be set or old ones to be overwritten or deleted. In some cases, even the activity of scrolling will result in a continuous score. All the sounds used originate from the *JavaScript* library for objective sound programming called timbre.js created by mohayonao[45]. The created sounds have very different qualities; some are softer,

[45] https://github.com/mohayonao

more harmonic or more noise-like than others, some blend better into the background etc. The sounds overlap so that there is a stacking process at some points of the user's interaction similar to the way that images are stacked in the *Hiperlook* browser. This might be increased by opening multiple tabs. Although the source code seems to suggest that only the active tab's cookies activity are examined, the sounds once initiated will not stop by switching to a different tab. The tabs, included to facilitate efficient parallel browsing[46], also function as natural separation between different web pages as they cannot be seen at the same time. The sound – being timed independently from the user's actions of potentially switching to a different tab – will bridge this separation. The enormous score created by a website like *expedia.com*, will still be perceivable if the user switches to a tab that uses cookies much more discreetly.

The differences in scores created by different websites seem to challenge users to seek out and compare examples; the noisiest, the quietest or most beautiful cookie-composition or even test preconceptions on the cookie policy of specific websites. The information on the web pages themselves is shown in the intended form and with its original content. However, the extension does not seem to invite users to dwell on pages and read the text or look at other visual elements. Instead, it leads them to listen to the initial sound composition, maybe to experiment a little with the mouse movement or scrolling and then turn to the next auditive experience. That is partly due to the initial volume-setting, set on maximum in the menu, which creates a rather noisy environment if the computer sound is on a normal level. In addition, the nature of the sounds is not altogether suitable for focusing on other sensory impressions. The path that users choose to navigate might differ significantly from that taken during normal browsing, being led more by curiosity regarding the sound-composition of a specific web page than by following content related information. Although the programme is an extension of a modern browser, it seems like the very prominent use of search engines is not what the programme suggests.

As previously mentioned, the extension inherits all layers of the conventional browser. However, it not only seems to broaden the information layer, as described above. It also seems to have an effect on the other lay-

[46] J. Huang, R.W. White, *op. cit.*

ers as well. Sound as a fundamentally time-based medium will affect the user's perception of time, especially as the cookies are audible. The sounds result not only from their mere existence but also from the processes of them being inserted, overwritten and deleted at specific times, notably at the beginning of the user's visit to a web page. The extension rhythmises the experience of the web page and adds different foci. Although it has no obvious influence on the visual layer of the browser, it seems to shift attention away from the displayed content information to a more scrutinising observation of the web page. This also influences the way that the user perceives the network. The bubbling of more technical aspects to the surface not only interrupts the continuous flow of content information, but also increases the perceived distance between the discrete websites. The user's behaviour might resemble more that of a "poking around", examining a concept of the Web as it already exists in the user's mind, looking at known websites but with the intention of gaining the information that is provided only by sound.

Internet conceptions and user roles

Summing up, we found a wealth of features and indications that amount to different conceptions of the Internet and the user. Let us briefly recapitulate the most evident results of our observations.

Internet conceptions. *Web Stalker*'s Internet is a set of interrelated connections that emphasises links. Web pages are not monoliths, but have multiple 'faces' that can be looked at separately with a variety of functional lenses in separate windows. This network has the potential to comprehensively spread along a topical logic as someone has set the links it follows. The contrary is the case with *.com.mx* where the domain name predefines the pool from which the web pages are fished, subordinating everything else (e.g. coherence in search topics). They are searched by using a formal parameter: the shortest URLs that can be produced with the according domain. When it comes to displaying the web pages, the directive seems to be 'back to the roots', i.e. the source code and text-only. Furthermore, the Internet is presented as something dynamic and performative by visually and acoustically emphasising the searching, connecting and loading processes. *Hiperlook 1.0* renders websites as isolated islands of added assets in a metric, yet seemingly endless space. These islands are not connected

with each other, but with the user (who entered the corresponding URLs). The fact that they remain closely connected to the user-cursor renders this situation an opportunity to check one's temper. *e-poltergeist* shows the Internet as a complex configuration where lots of information bits can be retrieved in different locations and as something dominated by alien rules, with an insistent attitude and recurring events. It brings something onto the user's desktop without them having requested it and thus shares aspects with "push technology". *Listening Back* reveals operative layers that users are rarely aware of in their everyday use. It may result in them emphasising their own established personal networks or make them curious to listen to the specific soundscapes of the best known websites.

User roles. The artistic browsers presented here all privilege some roles that can be assumed by their users. Our list is by no means exhaustive and aims to render the nuances more vivid, not to establish any rigid attributions. As for these artworks, no user studies exist yet. Methodologically, we are aware of falling short of ensuring adequate diversity when it comes to determining what "the" user is motivated to do in face of a specific design element. We discuss either our personal experiences (at least the four of us have different disciplinary backgrounds), or simplify by imagining a generic "user", while actually seeking to include a variety of different users and reactions.

Web Stalker offers an empty field without any activity. The user needs to make a first mark, then define a function for this distinction and refract a website in different windows to gather bits and pieces of information about it and to appreciate the "X-ray" map view of the link-skeleton. We might say it fosters a creative-proactive and scientific attitude. *.com.mx* drives users to extremes. They can either fight hard for the websites they actually want to see or fall into passivity or contemplation as the browser works incessantly and independently of an active user. It does its job alone. Those who see the browser auto pilot as a fascinating phenomenon will either start comparative experiments with the five compound 'specimens' or will accept it as co-creator in their own painting activity. *Hiperlook 1.0* styles the user as a conductor with a powerful single pointer for dealing with spatio-dynamical complexity. The task of rearranging the collage-web pages is challenging and can sort the users into tinkerers and meditators. Like *.com.mx*, *e-poltergeist* is provocative, generating fighters or defenders of their rights. With its hyperactivity on many fronts, the browser disarms the user who involuntarily becomes ineffective as a rescuer

hearing cries for help. Moreover, the laid out traces activate users' forensic instincts. *Listening Back* provides the user with a sensing toolkit for probing, testing and assessing a hidden dynamic para-network of machine activities. The user is inclined to investigate the tracking activity underlying familiar websites – which is where it potentially becomes personal.

Comparison with conventional browser features

Having recapitulated our findings directly deduced from the artworks, in this final section we want to include some observations from browser usage studies to introduce further parameters prominent in conventional browsing that help to highlight the specificities of how Internet use and access were envisioned in the browsers.

Hyperlinks and back buttons. While studies from the mid-1990s have determined that hyperlinks and back buttons are the most commonly used navigation mechanisms[47], none of these navigation habits is satisfied by the presented artistic browsers.

Tabs. Studies have shown that tabbed browsing alters users' strategies. For instance, users tend to work with tabs to browse in parallel in order to save time and ensure backtracking opportunities[48]. In *e-poltergeist*, the only artistic browser where tabs play a central role, the forced and automated tab generation leads to the exact opposite: the ordering principle for the user is rendered as ineffective as possible, the user's own browsing history is obscured, time is curbed.

Saving browsing history. Tabs are also used to keep a certain version of a web page. Thus, saving web pages seems to be a major concern[49]. Whereas *Web Stalker* offers an explicit saving functionality for most of the windows and their connected functions, even making it possible to access the data offline, the other art browsers do not provide intrinsic mechanisms to save data or created compositions. The fixed search phrases in *e-poltergeist* follow a programmed screenplay that creates a performative conservation, leaving the answer to the calls dependent on the fleeting web trends of the moment.

[47] Cf. P. Dubroy, R.Balakrishnan, *op cit*., p. 673.
[48] J. Huang, R.W. White.
[49] Cf. P. Dubroy, R. Balakrishnan, *op. cit*., p. 681.

Scrolling. Studies have found that "configure" tasks are the most common class of events in browsing, the majority of which are identified as scrolling[50]. Interestingly, most of the artistic browsers use scrolling to either show the compiled web page text or its source code in a smaller frame. However, it never seems to be at the centre of attention because before you can scroll you need to catch the frames in the first place – and these frames tend to float (*Hiperlook*), jump (*.com.mx*), duck (*e-poltergeist*) or be defunctionalised (*Listening Back*). Scrolling as the main activity for retrieving information in the user's conventional browsing behaviour highlights the extent of these evasive manoeuvres that open up the field for a variety of alternative interactions.

Conclusion

We conclude that artists do indeed open up alternative forms of accessing the Internet. With the conventional browser, we find a utility software that apparently gives a neutral view on the Web. As we were able to show, the artistic browsers on the other hand provide different foci and boldly insert specific topics, statements and 'colourations' as a lens to the Web. They create a narrow spectrum of possibilities that force the user to go through an embodied, channelled direction of action. This bias by design makes legible the actions imposed by conventional browsers and renders their enforcement tangible and thus accessible for reflection.

Acknowledgement

This research was funded by the German Science Foundation (DFG) within the academic framework of the Priority Programme "The Digital Image" (2019-2022) that pursues an open access policy.

[50] M.D. Byrne, and others, *The tangled Web We Wove: A Taskonomy of WWW Use*, "CHI, 1999", pp. 544-551, https://dl.acm.org/doi/pdf/10.1145/302979.303154, pp. 547-548.

Biography

Inge Hinterwaldner studied at the University of Innsbruck, and in 2009 she received her Ph.D in art history from the University of Basel, with a thesis on interactive computer simulations (*The Systemic Image*, German: Fink 2010, English: MIT Press 2017). Hinterwaldner taught at the universities of Berne, Basel, Lucerne, Lüneburg, and Zurich before fellowships and grants allowed her to pursue her research at MECS in Lüneburg (2014), Duke University in Durham (2015), and MIT in Cambridge/MA (2016). After teaching at Humboldt University of Berlin, in October 2018 she accepted a professorship for art history at Karlsruhe Institute of Technology (KIT). Her research focuses on interactivity and temporality in the arts, computer-based art and architecture, image and model theory, and the interdependence between the arts and the sciences since the 19th century.

Daniela Hönigsberg is an academic fellow at the Karlsruhe University of Technology in the Department of Art History. Since 2014, she has been doing her doctorate at the Technical University of Berlin on the topic of software as an artistic material. In 2015, she worked as a research assistant at the Technical University of Berlin for six months on the preparation of the proposal for the project Authorship 2.0. Her research focuses on computer-based art of the 1990s and early 2000s, interconnections of arts and sciences, and the theory of classical and early post-war modernism.

Konstantin Mitrokhov is a researcher at KIT in the Institute for History of Art and Architecture. His academic background is in art praxis, photographic arts, and systems analysis. Konstantin's research is focused on open-ended modes of knowledge production enabled by computation, spanning the fields of software studies, STS, film and media studies. He also works collaboratively as a video editor and cinematographer.

Martina Richter is a research assistant at the Karlsruhe Institute of Technology in the Institute for History of Art and Architecture. After studying computer science at KIT, she was e.g. working as a research assistant at Fraunhofer IOSB and as a guest artist at the ZKM | Center for Art and Media. The focus of her work is on robotics from different perspectives – in industrial, artistic and teaching fields.

Bibliography

Berners-Lee T., Cailliau, R., *WorldWideWeb: Proposal for a HyperText Project*, CERN, 12 November 1990, https://www.w3.org/Proposal.html, [accessed 7 June 2021].

Id., *Hypertext Markup Language – 2.0*, November, 1995, https://www.rfc-editor.org/rfc/pdfrfc/rfc1866.txt.pdf [accessed 14 June 2021].

Berners-Lee T., *Information Management: A Proposal*, CERN, May, 1990, https://cds.cern.ch/record/369245/files/dd-89-001.pdf [accessed 14 June 2021].

Byrne M.D., et al, *The tangled Web We Wove: A Taskonomy of WWW Use*, "CHI, 1999, 544-551", https://dl.acm.org/doi/pdf/10.1145/302979.303154, p. 550.

Catarci T., Costabile M.F., Matera M., *Visual metaphors for interacting with databases*, "SIG-CHI Bulletin", vol. 27, no. 2 (April 1995), pp. 15-17, https://doi.org/10.1145/202511.202514, p. 15.

Chang J.C., et al., *When the Tab Comes Due: Challenges in the Cost Structure of Browser Tab Usage,* "CHI Conference on Human Factors in Computing Systems", May, 2021, https://doi.org/10.1145/3411764.3445585, [accessed 12 July 2021], p. 3.

Clarke J.R., *WWW Page Metaphor Considered Harmful*, "HCI — A light into the future: Proceedings of OZCHI 95, 4th Australian conference on computer-human interaction", 1995, pp. 264-267, p. 264.

Dubroy P, Balakrishnan R., *A study of tabbed browsing among Mozilla Firefox users*, in "Proceedings of the SIGCHI Conference on Human Factors in Computing Systems", April 2010, pp. 673-682 https://doi.org/10.1145/1753326.1753426, p. 133.

Fellenz C., Parkkinen J., Shubin H., *Resolving conflicts between the desktop and the Web: a CHI 98 Workshop*, "SIG-CHI Bulletin", vol. 31, no. 1, January 1999, 26-28, https://doi.org/10.1145/329671.329684, p. 27.

Fuller M., Pope S., Green C., *I/O/D 4 The Web Stalker Version 1.0 for Windows 95 Help document (accompanying the executable of Web Stalker)*, 27 November 1997, https://bak.spc.org/iod/iod4.html [accessed 2 August 2021].

Gaffin A., *Take a walk on the Web site*, in "NetworkWorld: The Newsweekly of Enterprise Network Computing", 8 May 1995, pp. 57-59, p. 59.

Gibson J.J.., *The Senses Considered as Perceptual Systems*, Allen and Unwin, London, 1966; cf. [Anon.], *What is Skeuomorphism?*, "Interaction Design Foundation", (n.d.). https://github.com/mohayonao.

Huang J., White R.W., *Parallel Browsing Behavior on the Web*, in "HT '10: Proceedings of the 21st ACM conference on Hypertext and hypermedia", June 2010, https://jeffhuang.com/papers/ParallelBrowsing_HT10.pdf [accessed 20 July 2021], p. 4.

Kay A., *Computer Software*, in "Scientific American", September 1984, pp. 53-59.

Lederhaas S., Weidmann K.-H., *Webster: A New Information System for the Web*, in C. Stephanidis (ed.), *Proceedings of HCII 2014: Posters, Part I,* Springer, New York, 2014, pp. 180-185.

Norman D. A., Draper S.W., *User Centred System Design*, Lawrence Erlbaum, Hillsdale, 1986, esp. p. 46.

Norman D.A., *Affordances, Conventions, and Design*, in "Interactions", May-June 1999, pp. 38-42.

Id.., *The Design of Everyday Things*, Basic Books, New York, 2013, p. 159.
O'Reilly T., *Web 2.0: Compact Definition?*, "O'Reilly Radar", 1 October 2005, http://radar.oreilly.com/2005/10/web-20-compact-definition.html [accessed 30 July 2021].
Orgera S., *How to Turn on Opera Turbo Mode*, in "Lifewire", 17 February 2021, https://www.lifewire.com/activate-turbo-mode-in-opera-for-linux-mac-and-windows-4103691 [accessed 31 July 2021].
Post C., Golden P., Shaw R., *Never the Same Stream: netomat, XLink, and Metaphors of Web Documents*, in "DocEng '18 Proceedings of the ACM Symposium on Document Engineering", August 2018, https://doi.org/10.1145/3209280.3209530 [accessed 5 August 2021].
Reis C., Gribble S.D., *Isolating web programs in modern browser architectures*, "EuroSys '08: Proceedings of the 4th ACM European conference on Computer systems", April 2009, pp. 219-232. https://dl.acm.org/doi/10.1145/1519065.1519090, p. 219.
Rodriguez K., *BookLink browser to become freeware. Move follows Mosaic's lead*, in "InfoWorld: The Voice of Personal Computing in the Enterprise", 7 November 1994, p. 56.
Skjulstad S., Morrison A., *Movement in the interface*, in "Computers and Composition", no. 22, 2005, pp. 413-433, p. 416.
Snyder P., Ansari L., Taylor C., Kanich C., *Browser Feature Usage on the Modern Web*, in "Proceedings of Internet Measurement Conference", November 2016, pp. 97-110, https://doi.org/10.1145/2987443.2987466 [last accessed 12 July 2021], p. 97.
Tauscher L., Greenberg S., *How people revisit web pages: empirical findings and implications for the design of history systems*, "International Journal of Human-Computer Studies", no. 47, 1997, 97-137.
Tuvich M., *The Circular Page: Designing a Theatre of Choice*, in "Interactive Dramaturgies", 2004, pp. 221-229, doi:10.1007/978-3-642-18663-9_19.
Voigts R., Christmann S., Hagenhoff S., *Mobile Web Browsers*, University of Göttingen, 2011, http://webdoc.sub.gwdg.de/ebook/serien/lm/arbeitsberichte_anwebus/2011_01.pdf, [accessed 2 August 2021].
Yamaguchi T., Hosomi I., Miyashita T., *WebStage: an active media enhanced World Wide Web browser*, in "CHI '97: Proceedings of the ACM SIGCHI Conference on Human factors in computing systems", March, 1997, pp. 391-398 <https://doi.org/10.1145/258549.258811>.

Figures, figure captions and picture credit

Fig. 1: *WorldWideWeb* browser (1992) rebuilt as a web application, 2019. Screenshot.
Source: CERN 2019 WorldWideWeb Rebuild website, in: https://worldwideweb.cern.ch (2.8.2021). Screenshot by the authors.
Picture credit:

Fig. 2: Pei-Yuan Wei, *ViolaWWW Hypermedia Browser*, 1992. Screenshot version 1993.3.7, c. 1993.
Source: Viola Web Browser website, in: http://www.viola.org/ (2.8.2021).
Picture credit: wei@viola.org

Fig. 3: I/O/D (alias Matthew Fuller, Simon Pope, Colin Green), *The Web Stalker*, 1997-1998. Screenshot of the reprogrammed version from 2017.
Source: Screenshot by the authors via the emulator of the Rhizome's Net Art Anthology: https://sites.rhizome.org/anthology/webstalker.html
Picture credit: With kind permission by the artists.

Fig. 4: JODI (alias Joan Heemskerk and Dirk Paesmans), *.com.mx* from *%WRONG Browser* series, 2000. Screenshot showing the collaboration: While humans create continuous, fluid curves, the algorithm wangles, jumps and produces a visual staccato.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artists.

Fig. 5: Depiction of five composite brushes and the user's painting opportunities via the mouse movement in JODI's *.com.mx*. Detail outlines depict the underlying structure of the 'brushes'.
Source and picture credit: Konstantin Mitrokhov.

Fig. 6: Hernando Barragán and Andrés Burbano, *Hiperlook 1.0*, 2002. Screenshot.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artists.

Fig. 7: Mapping of *Hiperlook*'s viewport and mouse cursor indicator onto its non-oriented topology and the disjunct floating layers that display the websites' bundles.
Source and picture credit: Konstantin Mitrokhov.

Fig. 8: Alison Craighead and Jon Thompson, *e-poltergeist*, 2001. Screenshot.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artists.

Fig. 9: Jasmine Guffond, *Listening Back*, 2019. Screenshot showing the extension's tuning interface.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artist.

Finito di stampare
nel mese di maggio 2022
da Digital Team – Fano (PU)

Negotiating the way to the Internet

The impact of software design on the browsing experience and user interaction

Daniela Hönigsberg, Konstantin Mitrokhov, Martina Richter, Inge Hinterwaldner

Figures, figure captions and picture credit

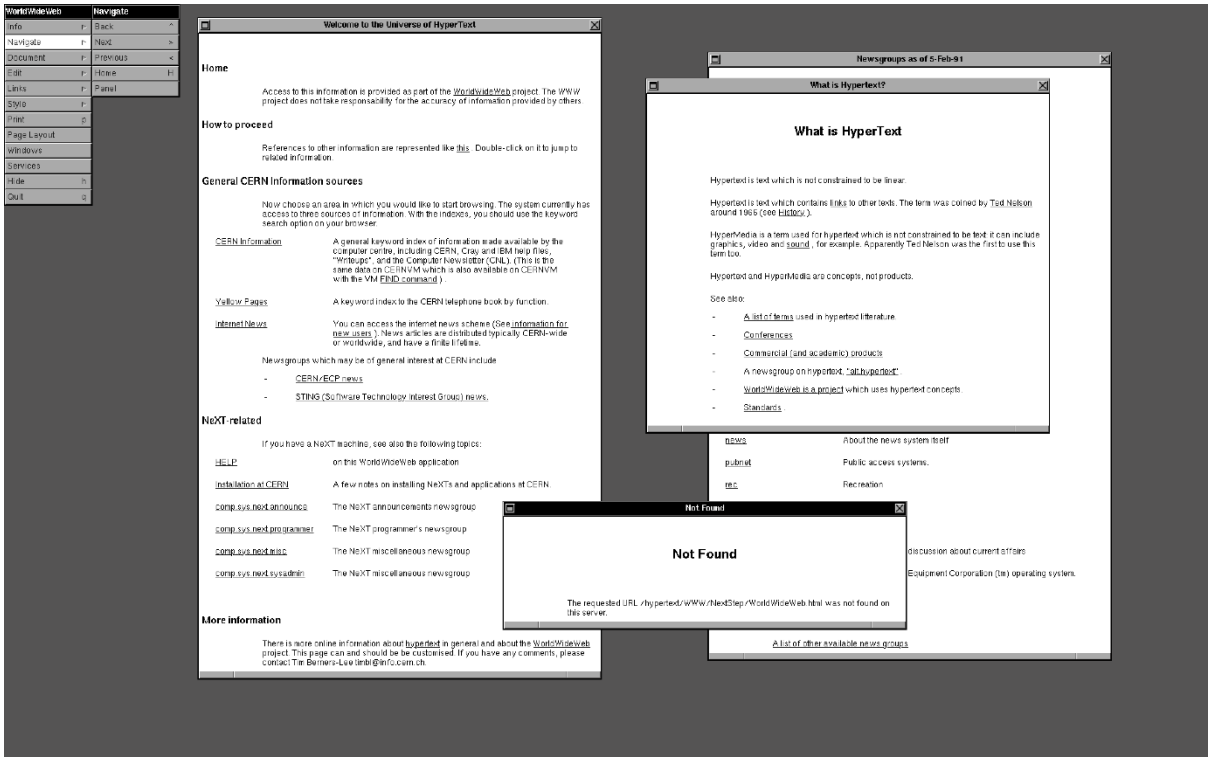

Fig. 1: *WorldWideWeb* browser (1992) rebuilt as a web application, 2019. Screenshot.
Source: CERN 2019 WorldWideWeb Rebuild website, in: https://worldwideweb.cern.ch (2.8.2021).
Screenshot by the authors.
Picture credit: CERN

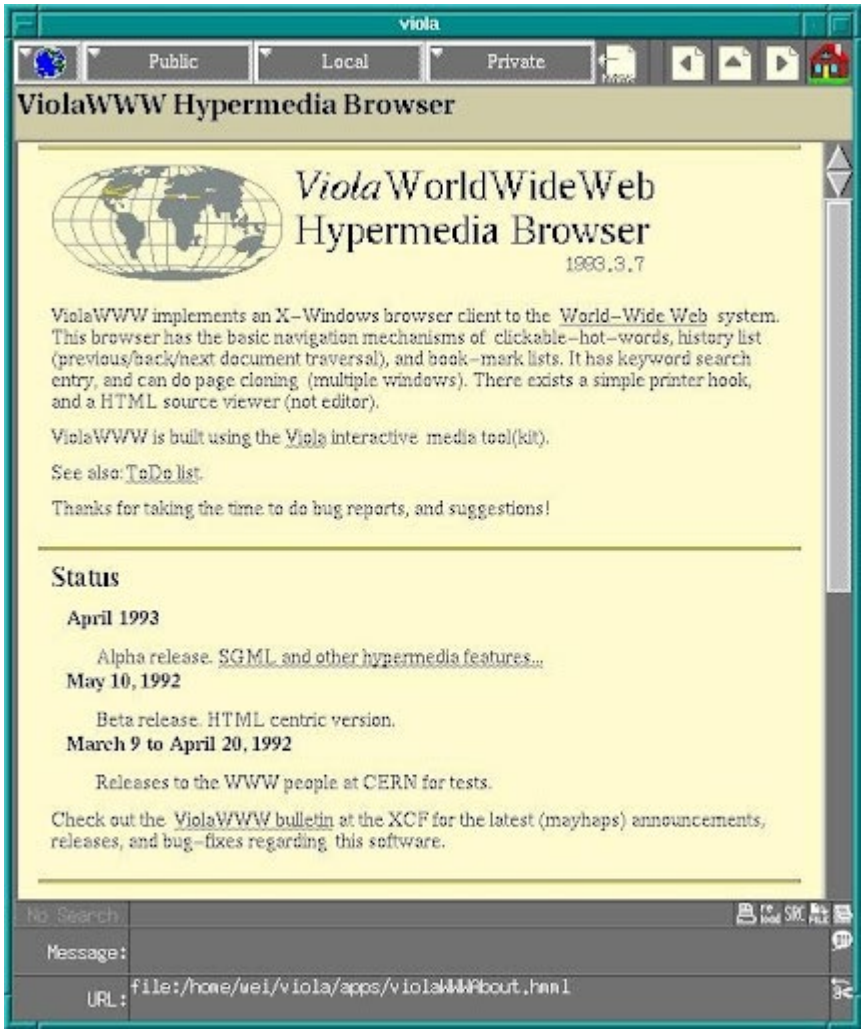

Fig. 2: Pei-Yuan Wei, *ViolaWWW Hypermedia Browser*, 1992. Screenshot version 1993.3.7, c. 1993.
Source: Viola Web Browser website, in: http://www.viola.org/ (2.8.2021).
Picture credit: Pei-Yuan Wei

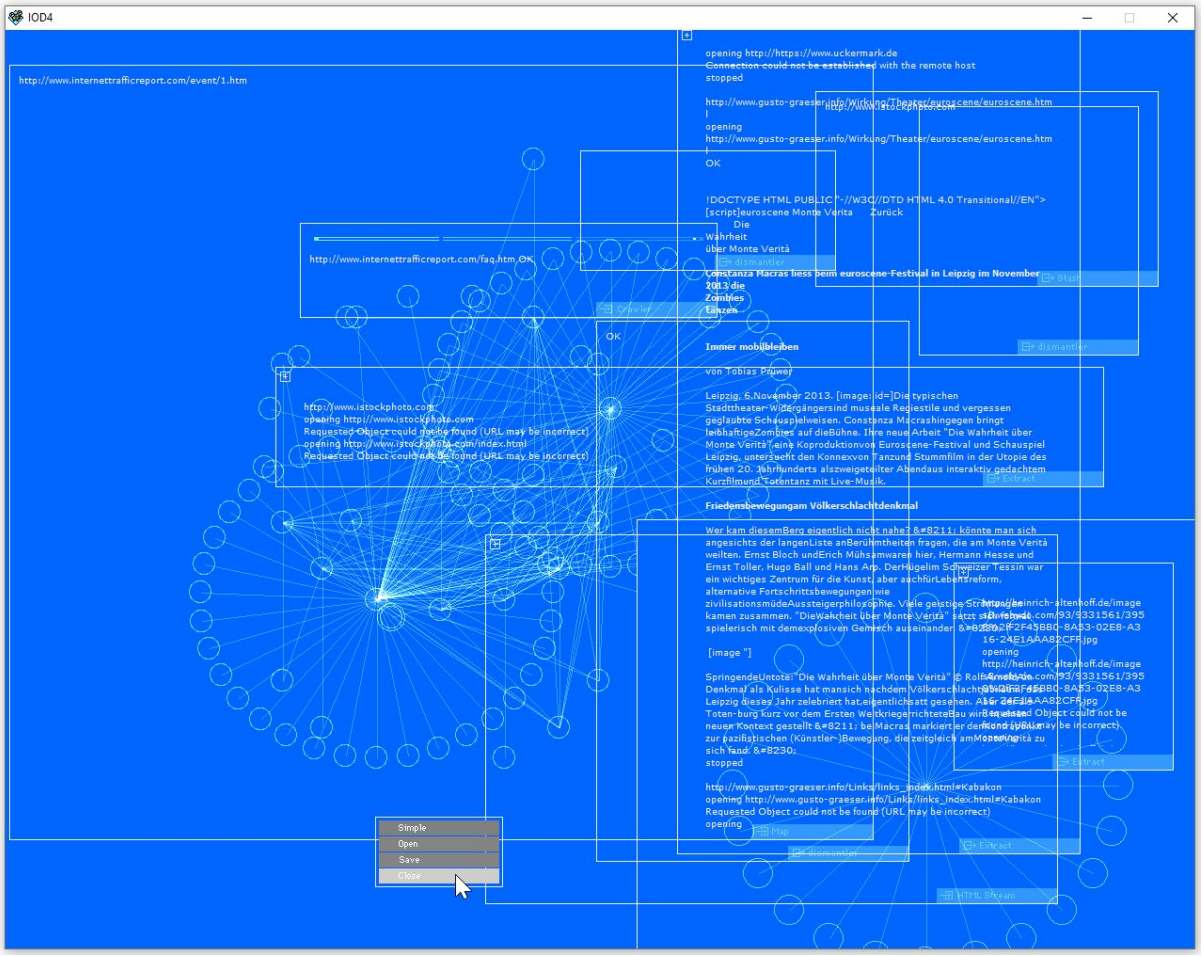

Fig. 3: I/O/D (alias Matthew Fuller, Simon Pope, Colin Green), *The Web Stalker*, 1997–1998. Screenshot of the reprogrammed version from 2017.
Source: Screenshot by the authors via the emulator of the Rhizome's Net Art Anthology: https://sites.rhizome.org/anthology/webstalker.html
Picture credit: With kind permission by the artists.

Fig. 4: JODI (alias Joan Heemskerk and Dirk Paesmans), *.com.mx* from *%WRONG Browser* series, 2000. Screenshot showing the collaboration: While humans create continuous, fluid curves, the algorithm wangles, jumps and produces a visual staccato.
Source: Screenshot by the authors. Picture credit: With kind permission by the artists.

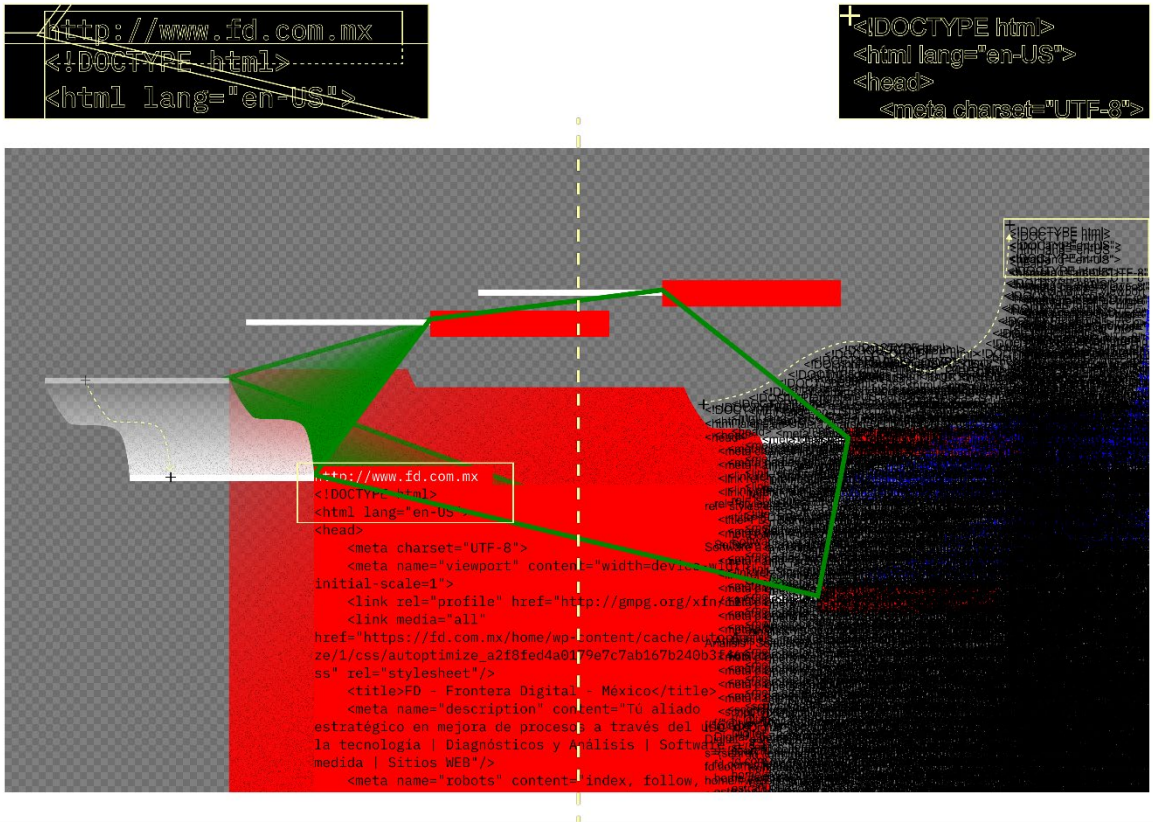

Fig. 5: Depiction of five composite brushes and the user's painting opportunities via the mouse movement in JODI's *.com.mx*. Detail outlines depict the underlying structure of the 'brushes'.
Source and picture credit: Konstantin Mitrokhov.

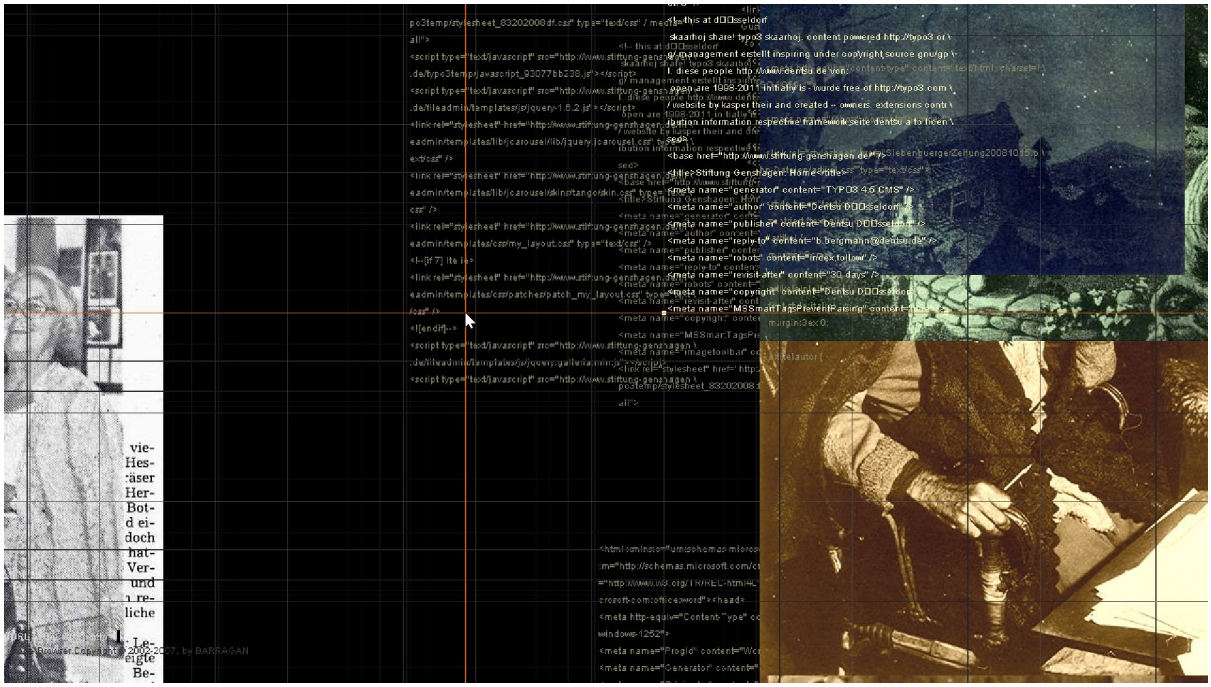

Fig. 6: Hernando Barragán and Andrés Burbano, *Hiperlook 1.0*, 2002. Screenshot.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artists.

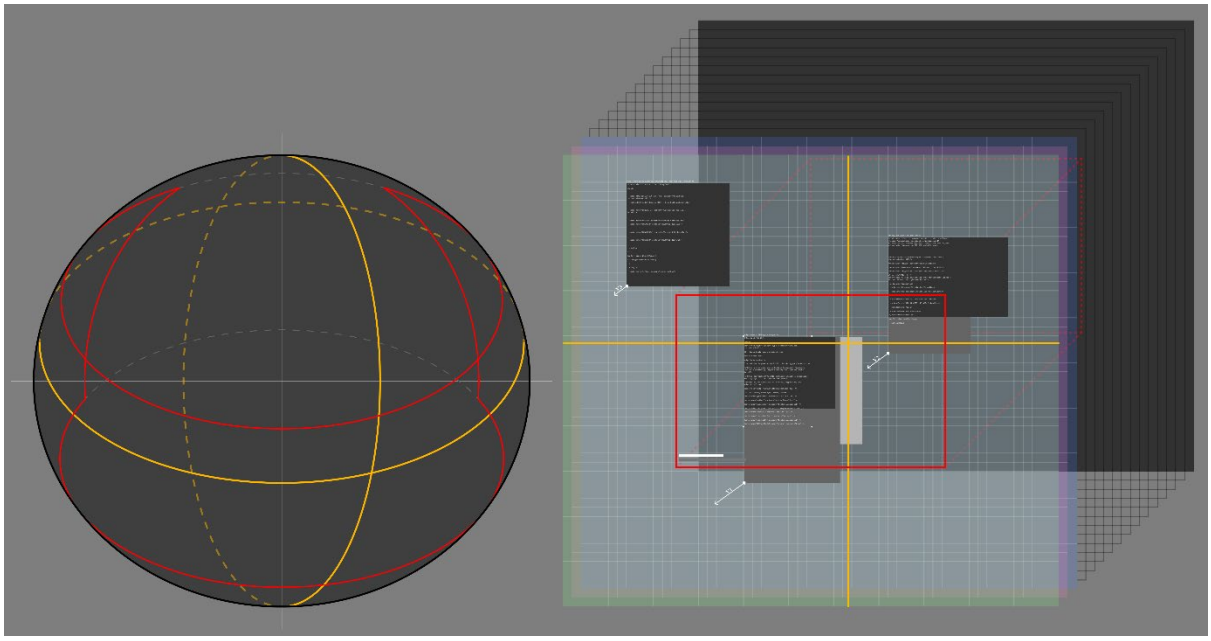

Fig. 7: Mapping of *Hiperlook*'s viewport and mouse cursor indicator onto its non-oriented topology and the disjunct floating layers that display the websites' bundles.
Source and picture credit: Konstantin Mitrokhov.

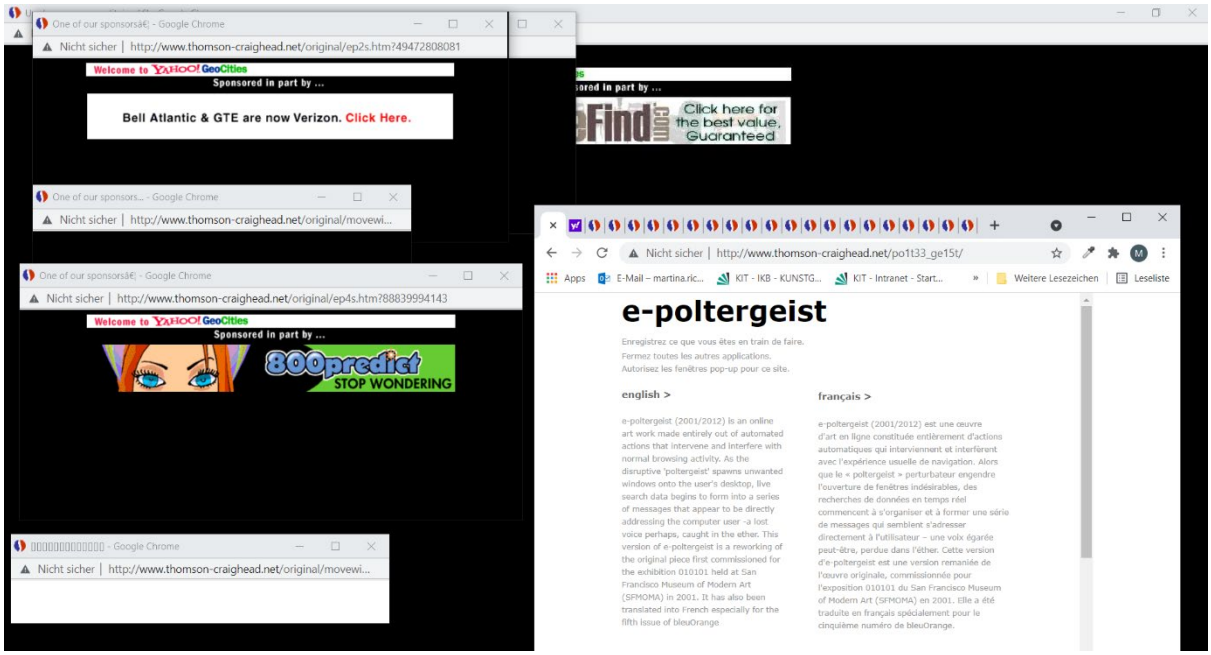

Fig. 8: Alison Craighead and Jon Thompson, *e-poltergeist*, 2001. Screenshot.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artists.

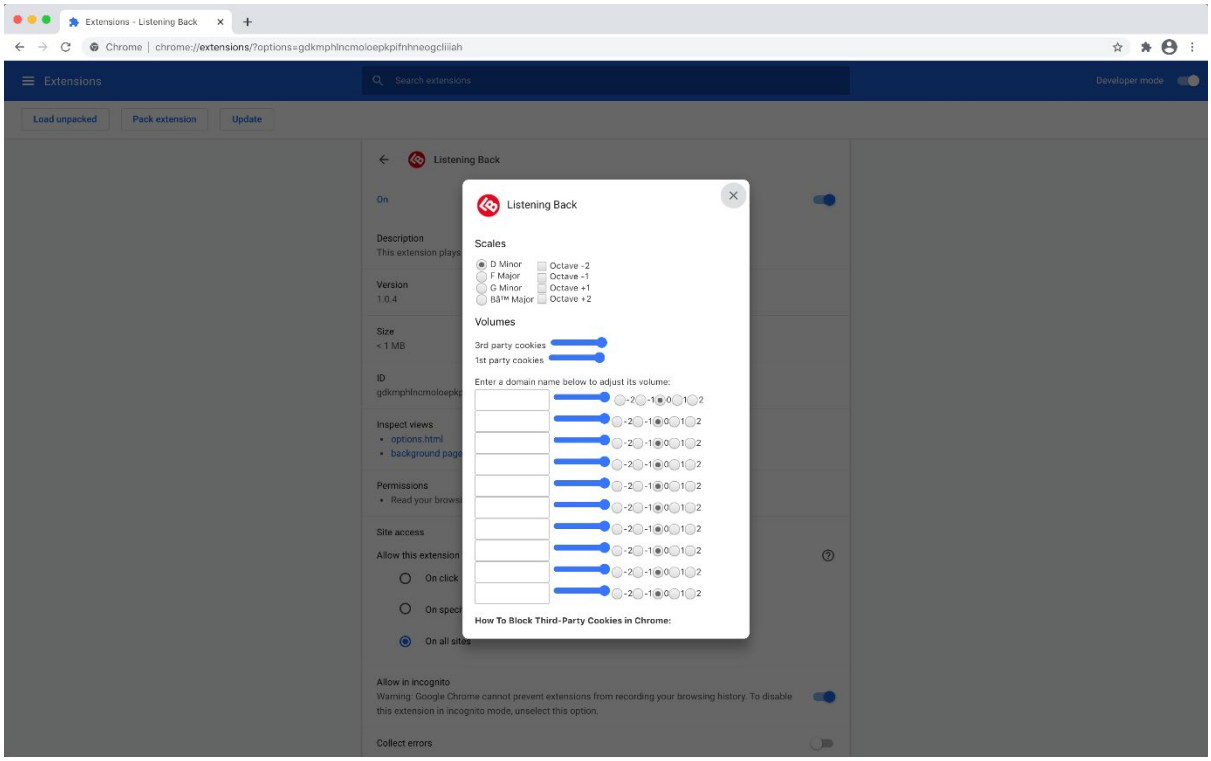

Fig. 9: Jasmine Guffond, *Listening Back*, 2019. Screenshot showing the extension's tuning interface.
Source: Screenshot by the authors.
Picture credit: With kind permission by the artist.