# Revisiting creative behaviour as an epistemic process: lessons from 12 computational artists & designers

Martin Tricaud, Michel Beaudouin-Lafon

## To cite this version:

Martin Tricaud, Michel Beaudouin-Lafon. Revisiting creative behaviour as an epistemic process: lessons from 12 computational artists & designers. OzCHI '23: Proceedings of the 35th Australian Conference on Human-Computer Interaction, Dec 2023, Wellington, New Zealand. 10.1145/3638380 . hal-04415624

HAL Id: hal-04415624
https://hal.science/hal-04415624v1

# Revisiting creative behaviour as an epistemic process: lessons from 12 computational artists & designers

Martin Tricaud
Michel Beaudouin-Lafon
martin.tricaud@lisn.upsaclay.fr
mbl@lisn.fr
Laboratoire Interdisciplinaire des Sciences du Numérique
Université Paris-Saclay, CNRS, Inria
Orsay, France

## ABSTRACT

In this article, we investigate creative behavior among computational artists and designers, in order to improve our understanding of the interaction mechanisms that they rely on to identify and appropriate the mediating properties of code and computational representations. We conducted an observational study with 12 computational artists and designers working with visual media. The results lead us to analyze creative behavior as an epistemic process, whereby agents generate knowledge about their medium through *epistemic actions*, and produce their medium by externalizing this knowledge into *epistemic artifacts*. We discuss the implications of these findings for the design and evaluation of interactive systems for creativity.

## CCS CONCEPTS

• **Human-centered computing → HCI theory, concepts and models**; **Empirical studies in HCI**;

## KEYWORDS

creative behaviour, instrumental interaction, epistemic action, epistemic artifact, procedural computer graphics, digital arts, creative coding, generative design

**ACM Reference Format:**

Martin Tricaud and Michel Beaudouin-Lafon. 2023. Revisiting creative behaviour as an epistemic process: lessons from 12 computational artists & designers. In *OzCHI 2023 (OzCHI 2023), December 2–6, 2023, Wellington, New Zealand.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3638380.3638395

## 1 INTRODUCTION

Creative users often use software in unexpected ways. They act as "lead users" [103] by pushing the limits of technology and finding new possibilities for action in their environment. The advent of computer-based creativity-support tools has expanded the scope of creative practices to embrace computation itself as a medium.

This paper seeks to deepen the HCI community's understanding of artistic and designerly practices in which computing is the predominant medium of creation, such as creative coding, computer-generated imagery (CGI), parametric and generative design, or visual effects. Our motivation is to understand the mechanisms of interaction that drive the emergence of the creative medium — that is, the processes artists and designers rely on to recognize that code, data and computational representations in general have mediating properties that are relevant for producing value and novelty.

In that regard, the position we adopt regarding the concept of creativity aligns with the framework of 4E (Embodied, Embedded, Enacted, Extended) cognition [63]: We hypothesize that the ability of an agent to generate novelty and value, which is commonly regarded as a definition of creativity [10, 68, 94, 105], is not solely a product of individual mental capacity, but also influenced by a set of material conditions that predispose the environment to become a medium for creation.

The core contribution of this work is partly empirical and partly theoretical. After a review of related work, we report on an interview study of creative behavior among 12 computational artists and designers, which we structure along two dimensions. First, we find that an essential driver of creative behavior is the ability to engage in what Kirsh & Maglio call *epistemic actions* [52], i.e. actions that are not intended to bring the agent closer to a goal, but to probe the task environment to uncover further possibilities of action, and we detail how participants rely on these actions to form new knowledge about their medium. Then, we introduce the concept of *epistemic artifact* to capture the notion that participants externalize the knowledge derived from epistemic actions into objects that enrich the medium with new possibilities for action. We conclude with a discussion of the implications of these findings for the design and evaluation of interactive systems for creativity.

---

*OzCHI 2023, December 2–6, 2023, Wellington, New Zealand*

The definitive Version of Record was published in *OzCHI 2023 (OzCHI 2023), December 2–6, 2023, Wellington, New Zealand*, https://doi.org/10.1145/3638380.3638395.

## 2 RELATED WORK

In an extensive literature survey from 2018, Frich et al. [30] observed that the concept of creativity is often vaguely defined in HCI, if at all. Yet how one analyses and defines creativity can vary significantly depending on what we are trying to do with this definition, and has a strong influence on how others will leverage the concept. To ground our research, we sought to understand the motivations that underlie different views on creativity, and how they inform the design of interactive systems.

### 2.1 Definitions of creativity and their technical legacy

Framing creativity as one of the milestones of artificial intelligence — as proposed by the fathers of symbolic AI [60] — inevitably calls for a symbolic, mathematical and computationally tractable definition of the concept. Herbert Simon contributed fundamentally to this effort, first by framing design as a search for satisfying solutions in an ill-defined problem space [86], and perhaps more importantly by equating creativity with a sophisticated form of design [65, 85], in which defining the problem is a part of the problem-solving process, and where the fitness of solution is assessed according to some measure of novelty. Simon's works were key to developing the design methods field, whose members believed that capturing design expertise into more or less formalized conceptual frameworks could contribute to rationalizing innovation processes in organizations [35, 36].

Schoen criticized Simon's approach for overlooking the physically situated dimension of design, which he describes as *thinking in action*. He emphasizes the notion that the materials of a design situation shape the cognitive process of creative designing [83]. Malafouris' development of Material Engagement Theory [55] follows a very similar intuition. He argues that the boundaries of what we call the *mind* fluctuate depending on how one engages with the materiality of the environment [56], and builds on this framework to discuss the applicability the the notion of materiality to digital environments in creative activities [74]. Oxman, an architect and researcher, derives comparable conclusions from her practice, observing that *"in new digital workflows, creative practitioners interact with, control and moderate generative and performative processes and mechanisms"* [72] and concludes that information has become a new *material* for the designer, which mandates a re-examination of creative practices.

Advances in interactive evolutionary computing [28], and more recently in deep learning and generative models have produced new tools and design workflows where the role of software shifts away from tool to partner in the creative process [26]. While some anticipate that *human-in-the-loop* approaches will simply displace the skills of creativity from manual to conversational [18, 69], creative professionals such as Burry have expressed concerns about technologies that deprive designers of their agency [19].

Our position is that defining creativity through some qualities of its final outputs, i.e. *novelty* and *value*, is bound to yield tools that will try to optimize for that quality, i.e. that provide shortcuts towards a pre-conceived goal and ultimately deprive designers, artists and software users in general of their agency — as Burry fears. Instead, one should look to characterize creative processes not through qualities of their final outputs, but through qualities of the process itself and of the material conditions in which it takes place. This is the approach we take in the present work.

### 2.2 Computing as a creative medium

A fundamental feature of computers is that they enable operations unavailable to embodied human actions, that is, the automated production and transformation of symbolic representations [79]. Pioneers of the field recognized very early on the creative possibilities of combining computing with displays or any device suited for producing visual content from symbolic instructions [102].

At present, building and manipulating abstract symbolic representations has become a fundamental aspect in many creative practices, old and new: Creative coding and generative art [11] — popularized by Processing [76], has brought into the mainstream the idea that procedural and computational approaches can serve as a creative medium. More established fields such as architecture and industrial design have also embraced this paradigm, giving rise to new tools for parametric design [73], such as the Grasshopper [82] plugin for Rhino or the xGenerative Design editor in Catia [96]. The video game and CGI industry would not be what they are today without the development of procedural content generation techniques [38, 89] that keep expanding the amount and diversity of shapes, structures and effects that can be synthesized by computer. Despite having different motivations and constraints, software tools used in these communities share a large common ground of generative techniques such as shape grammars, L-systems, constructive solid geometry, noise, reaction-diffusion, shape grammars or flow fields.

In the HCI community, these developments have sparked new investigations of creative practices and a reflection on how software can support creative activities in a way that goes beyond the mere transposition of non-digital workflows into digital environments. In generative design software the designer's practices have shifted from modeling shapes to expressing designs as the parameterized pipeline of instructions that generate them [72]. Therefore, the program is not just the environment in which creative tasks occur, but a material that mediates creative tasks.

The practice of live-coding takes things further, endowing the act of authoring and modifying code with a performative value [8, 62]. This imposes additional requirements on underlying software architectures: The semantic correctness of changes introduced by the user must be automatically detected to avoid crashing the program during a performance [21, 61]. However aware artists may be of the expressive possibilities offered by code, retaining some form of tangible gestural expressivity is a recurring concern, not only in configuring the programs' inputs, but also in building the programs. For instance, Hook et al. [41, 42] have observed that live audiovisual performers also want to be able to reconfigure and produce their pipelines at runtime. A similar motivation underlies the design of the Reactable by Jorda et al. [50], an interface that lets performers modify the architecture of an audio-visual synthesis instrument on the fly through tangible interaction.

## 2.3 Reclaiming materiality in creativity support software

With the rise of computing as a medium for creative design, discussions on the meaning of materiality in the context of digital environments have become a central issue in HCI [33, 74]. Indeed, as explained by Bratteteig, *"The materials and tools we use in design influence which possibilities we see and choose to realize"* [15]. However, the challenge posed to HCI by computational arts and design is that there is no *a priori* physical metaphor that underlies the behavior of the *objects of interest* (i.e. code and symbolic abstractions) that constitute the medium of creation.

Designers' ability to identify and achieve future possibilities depends on how well they understand the *materials* [15], that is, the elements of the task environment that can be shaped, sensed, probed and assembled. In that sense, creative practices rely on cognitive predispositions that evolution has tailored for a material world. For instance, Hegarty has shown that mechanical reasoning enables humans to plan actions using mental simulation rather than by descriptive inference [37], and Osiurak et al. posit that reasoning about the physical properties of objects underlies our ability to use and create tools [71]. Torrents et al. propose that the manipulation of physical constraints modifies the exploratory behavior of individuals and enables novel affordances to be acted upon [100], while Banfield showed that embodied physicality of artists' activity enables a state of flow [3].

Can we reclaim some form of materiality in digital creativity support tools by making them more permeable to such cognitive predispositions while at the same time retaining the expressivity permitted by algorithmic procedures? This has been one of the concerns of HCI since the inception of the field, and a driving force behind the development of Direct Manipulation interfaces, starting with Sutherland's Sketchpad [95]. Among other approaches, Beaudouin-Lafon's instrumental interaction [4] is driven by the intuition that the materiality of software environments is conditioned by the availability of instruments to experience it. Decoupling instruments from the environments in which they operate enables users to adapt software to their needs by reusing these instruments in different contexts. Tangible interfaces [48] embody digital information into physical objects, thereby building on our natural skills for manipulating objects. Materiality and tangibility lend themselves well to embodied action, which plays a crucial role in creative behavior [23, 49].

In this article, we take a human-centered perspective on creativity to understand how artists and designers deal with the immaterial nature of software and abstract representations. To this end and unlike most previous work, we study artists and designers who embrace programming and generative procedures as part of their practice.

# 3 OBSERVATIONAL STUDY

## 3.1 Profiles and practices of the participants

We conducted a total of 14 interviews with 15 participants (the 7th interview was a joint interview with 2 participants who were close collaborators), which were recruited through various channels (see Table 1).

### 3.1.1 *Recruitment process.*

Our criteria for including an interview in our analysis were that the participant should have a creative practice that entails *building and/or configuring algorithmic procedures* to produce *visual works*, and be able to *provide detailed lived experiences* of their creative practice. "Building and/or configuring algorithmic procedures" is to be understood in a broader sense than simply *textual coding*, and encompasses any form of creative workflow in which the user's actions do not directly target the rendered output, but instead modify the procedure that generated it, as well as its input parameters. However, to avoid any confusion, we never used this phrasing when reaching out to potential participants. Instead, we provided a list of terms colloquially used by creative communities to establish a distinction with direct editing workflows — such as *generative design*, *parametric design*, *generative art*, *procedural content generation*, *computer generated images* — and asked the participants whether they felt their practice related to one or several of these terms.

Whether or not this criterion is met can sometimes be inferred ahead of the interviews from the type of environment participants report using: Software such as Houdini [92] or Grasshopper3D [82] are entirely geared towards the production of visual content through visual programming, and therefore users of such tools fall within the scope of the study. For users who worked with environments that target a broader range of use cases with multiple editing paradigms (parameterized non-destructive edits, direct editing and sculpting, visual programming), such as Blender [81] or Unity [99], a few framing questions allowed us to determine the extent to which they exploited the more programmatic and parametric aspects of visual creation within them.

Based on these requirements and in order to keep a focus on visual creation practices, we excluded **P5** from the analysis, since his work was exclusively in the area of music and sound design. We also discarded **P13** because the framing questions led us to consider that the building and configuration of algorithmic procedures were not central to her practice. Finally, we chose not to include **P14** because the discussion constantly drifted to very general and high-level considerations on arts and algorithms, and we were unable to steer the questions towards actual detailed accounts of situated experiences that are necessary for this methodology.

### 3.1.2 *Demographics, sampling, and subsequent limitations.*

The demographics of the remaining 12 artists shows that the participant had an average age of 32, with a 24-53 range and a standard deviation of 8.7. Participants, overwhelmingly identified as male (11M, 1F), which is arguably one of the limitations of our study, which we address in the discussion section.

| Participant | Country | Age | Gender | Recruitment channel | Training in computing | Situatedness |
|---|---|---|---|---|---|---|
| P1 | FR | 26 | M | Thematic communities | High | 2.5 |
| P2 | FR | 29 | M | Thematic communities | High | 3 |
| P3 | FR | 24 | M | Thematic communities | High | 2 |
| P4 | FR | 44 | M | Thematic communities | Little | 3 |
| P5 | FR | 39 | M | Thematic communities | Some | 3 |
| P6 | SK | 26 | M | Personal / pro network | High | 2.5 |
| P7a | UK | 24 | M | Personal / pro network | Some | 3 |
| P7b | NL | 25 | M | Personal / pro network | Some | 3 |
| P8 | PL | 36 | M | Direct inquiries | High | 2.5 |
| P9 | FR | 30 | M | Venues + events | Some | 3 |
| P10 | CA | 31 | M | Personal / pro network | Some | 3 |
| P11 | FR | 30 | M | Thematic communities | Some | 1.5 |
| P12 | FR | 53 | F | Venues + events | Some | 1.5 |
| P13 | FR | NA | F | Venues + events | Little | 1.5 |
| P14 | AR | NA | M | Personal / pro network | High | 0.5 |

**Table 1: Participants and recruitment channels. Participants on grey backgrounds were not included in the analyses (see text). Situatedness is scored as follows: interview took place in the participant's work environment (1 point), in person but not in the participant's work environment (0.5 point), remotely (0 point); the participant was able to showcase their tools, devices and processes (1 point), or did not have them at hand but was able to give a detailed account of them (0.5 point); the participant was able to showcase the output of their creative work (1 point).**

#### 3.1.3 *Participants background.*

Most participants (except **P4**, **P9** and **P10**) had received academic training in STEM and computing-related areas. In some cases, their educational background overlapped with or was completed by training in more creative or artistic fields, such as visual effects and CGI (**P1**, **P3**), digital media art (**P6**) or interaction design (**P8**).

Though not all of them were formally trained in programming and computer science, they all have a high level of digital literacy, i.e. they can at least all confidently use a high-level programming language for scripting (though the most skilled ones can build full-fledged software from scratch), and all of the projects they presented reflected an acute awareness of the creative possibilities that programming brings about. As participants often reported (**P1**, **P4**, **P6**, **P8**), this awareness owes a lot to community-sourced content, including open-sourced works of creative coding, tutorials and social media.

#### 3.1.4 *Overview of the creative practices of participants.*

As we expected based on our criteria for participation in the study, the creative process of all participants involved an important part of (a) procedural content creation, meaning that the participants always needed at some point to put together a pipeline of instructions either through text coding or visual programming languages, sometimes with a mix of both, and/or (b) parametric exploration, meaning that they needed to spend time adjusting the parameters of a pre-existing procedure.

Despite these similarities in the process, the nature of the works produced by the participants spanned a wide variety of forms. Some projects were live visual performances (**P1** — Fig. 1, **P2**, **P3**, **P4**, **P9**, **P12**), generative 3d sculpture and 3D printing (**P6** — Fig. 2, **P7**, **P10** — Fig. 13), architectural designs (**P11**), data-visualizations (**P8** — Fig. 12), video art, visual effects and motion design (**P1**, **P2**, **P3**, **P9** — Fig. 3, **P12** — Fig. 7) or 2D stills and prints (**P4** — Fig. 4, **P12**);

These projects included both commissioned design works (**P1**, **P3**, **P8**, **P11**) and personal work, as well as works that cannot be labeled as either: some projects were self-commissioned (**P7**), some were submissions to calls for artistic projects with a very open-ended brief that offered a lot of creative freedom (**P9**). Even participants who routinely applied their skills for client work relied heavily on personal projects to build and refine their creative know-how.

**Figure 1: MIDI-controlled interactive glitch-art system (P1)**

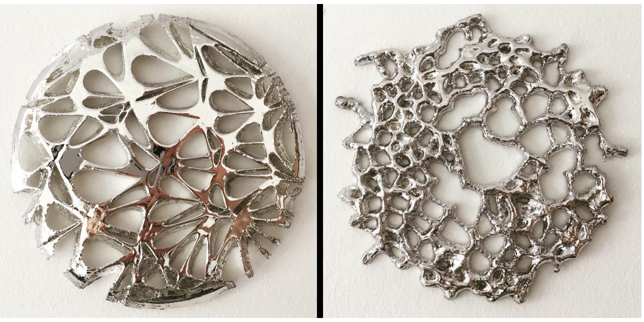


**Figure 2: 3D-printed generative sculptures designed on Blender (P6).** *Acheiropoieta.* © 2019 Jakub Fiala**.**

**Figure 3: Still from a video art piece designed for geodesic dome projection, made with Unity (P9).** *Apoptose.* © 2017 Sophie Le Meillour & Fabrice Starzinskas

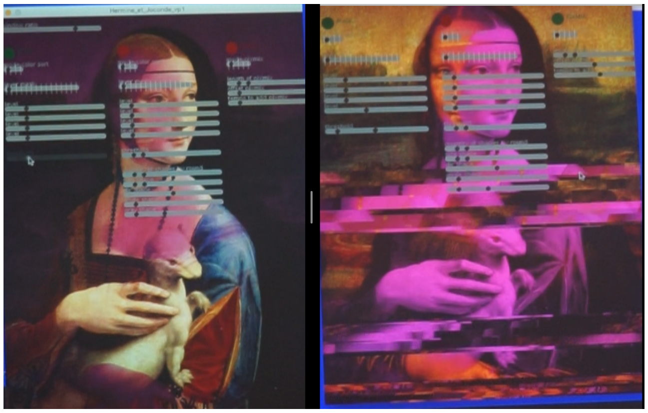


**Figure 4: System for procedural image composition built with Processing by P4, used for the series *La Joconde à l'Hermine*** [57]

## 3.2 Data collection and analysis

Interviews lasted from 40 to 80 minutes and were conducted face-to-face, at the participant's preferred location, except with **P11** where it was done via a video call, and **P12** where the first part of the interview was conducted face-to-face and the second part over a video call. The interviews usually started with the participant giving a brief overview of their portfolio and their creative practice. Then we focused on one (or more, if time allowed) specific project that the participant felt exemplified the challenges and opportunities of procedural creation. In all interviews except with **P11** and **P12**, participants were able to showcase the tools and processes that they rely on for their creative work.

The interviews followed an idiographic approach [75], i.e. they were not conducted with specific patterns to look out for in mind but with the goal of gathering personal accounts of real experiences. The participant was then asked to guide the interviewer through the different steps of their creative process, providing as much detail as possible, including recalling their goals and intentions, requirements and constraints, and specific problems and breakdowns encountered. They were also asked to reproduce the steps of a given task, when possible, and to highlight the workarounds and strategies deployed to address problematic situations. We paid particular attention to the expression of frustrations, excitement and opportunities by the interviewees and to the objects of interest involved in these experiences.

All interviews were audio- and video-recorded and then transcribed. They were then analyzed using the principles of Interpretative Phenomenological Analysis or IPA [90, 91]. This method bears several similarities with Braun & Clarke's reflexive Thematic Analysis [16, 17], but its use in Human-Computer Interaction is much more recent [1, 54]. A significant difference is that IPA concerns itself with making sense of the participant's own sense-making, which is particularly relevant here since artists who are skilled enough to build their own tools have inevitably reflected on them. As a consequence, the units of meaning used for coding tend to be of a higher semantic level than in Thematic Analysis. In other words, the data is interpreted before looking for patterns, whereas if we had used Thematic Analysis, the search for patterns would have taken place before the interpretation. Aggregating those codes into themes also followed the approach favored by IPA, in the sense that we interpreted the observations before looking for patterns across them.

## 3.3 Development of the analytical framework and its theoretical foundations

Defining the units of meaning — i.e. deciding whether an observation or a statement qualifies as relevant material for the analysis — is a process that is necessarily primed by our pre-existing understanding of the practice we study, both from a personal standpoint (one of the authors is a hobbyist visual artist and musician), and from a scientific or academic standpoint.

For instance, we knew both from personal experience and from Marks et al.'s seminal paper [58] that *slider tweaking*, i.e. moving a slider widget with either no clear understanding of how the slider maps to perceptual feedback or no precise knowledge of where the cursor should be positioned to achieve a desired result, is a frequent frustration in computer graphics. As a predictable consequence, instances of tweaking-induced frustrations were quite salient from

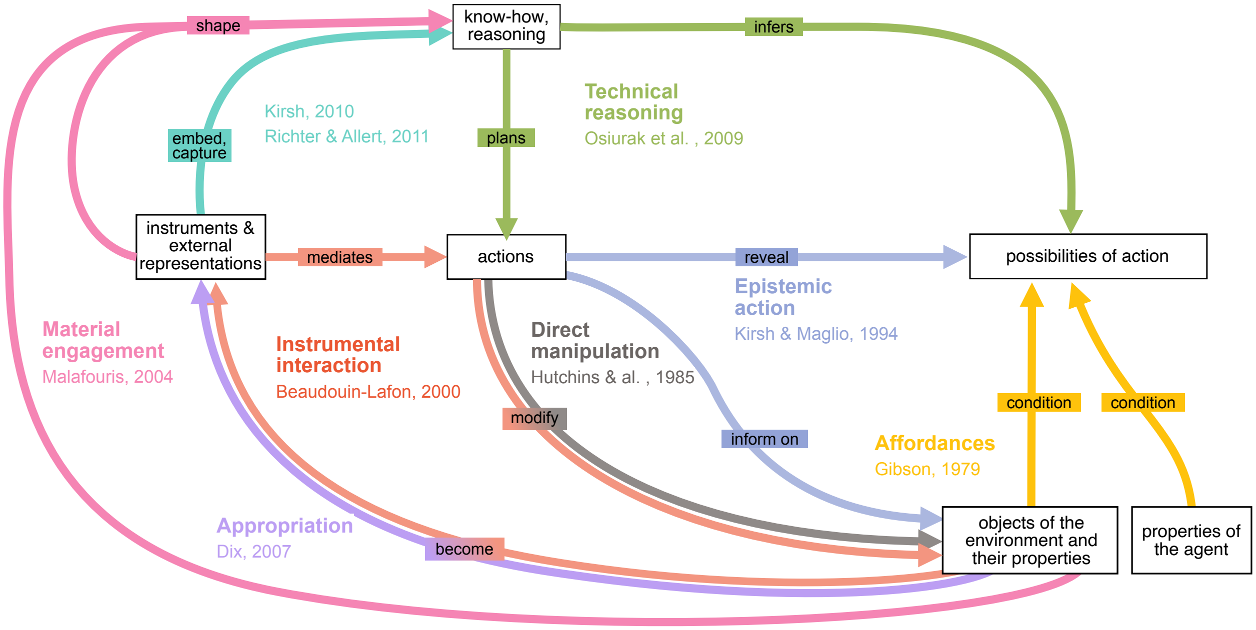


**Figure 5: Network of thematic connections between the different theories and frameworks that influenced the analysis**

the beginning. We were also able to recognize that slider tweaking was often used for exploring options rather than reaching a goal, which we identified as an instance of Epistemic Action [52], thanks to our prior familiarity with the concept and with other works in which it appears [53, 97].

The conceptual proximity between Kirsh & Maglio's notion of epistemic action [52] and Cadoz et al.'s work on the epistemic value of instrument-mediated gestures [22] led to an exploration of instrumentality, bringing in Beaudouin-Lafon's theory of instrumental interaction [4] and the neighboring concepts of affordances [32], direct manipulation [44], appropriation [27] and technical reasoning [70, 77] to generate new interpretations of the data. Finally, a number of theoretical reflections point towards concepts close to the notion of *epistemic artifact* that we introduce. For example, Kirsh [51] underlines the epistemic character of external representations beyond the role of container to which they are often reduced and shows that people frequently build representations for the explicit purpose of probing them. Similarly, Richter et al. [80] identify an epistemic role for artifacts, stating that *"artefacts in this sense are 'productive things' in that they are not limited to represent what exists but also to provide insight into what might or could be."* Fig. 5 illustrates the network of conceptual connections between the theories and frameworks that served as the seed for generating the themes of our analysis.

### 3.4 Results

The main finding of this study is that the *mediating properties* of the environment are both uncovered and produced. In the first part of the analysis, we argue that the process of discovering mediating properties corresponds to what Kirsh & Maglio [52] have called *epistemic actions*, i.e. actions performed by an agent to explore the possibilities of action offered by the current configuration of the software environment and the objects of interest it contains. The term is to be contrasted with pragmatic actions, which aim to achieve a pre-identified goal.

In the second part, we highlight the fact that users *produce* their medium by endowing their environment with new mediating properties. This is accomplished through the externalization of their knowledge into what we call *epistemic artifacts*, which make this knowledge persistent, shareable, and reusable.

## 4 EPISTEMIC ACTIONS

### 4.1 Speculative aspects of technical reasoning

In many of our observations, it appeared that many epistemic actions were driven by a speculative discourse. Whereas design concerns itself with "what ought to be" [87], the initial motivation behind many projects was often to uncover "what could be": *"What are the different ways in which I can distort a matrix of pixels?"* (**P1**, **P4**); *"sculpt and animate a point cloud?"* (**P1**, **P6**, **P7**, **P9**); *"visualise a dataset?"* (**P8**)

**Figure 6: Sculpting with a simulation fluid for the project *Algorithms Unmasked*(P7) [24]**

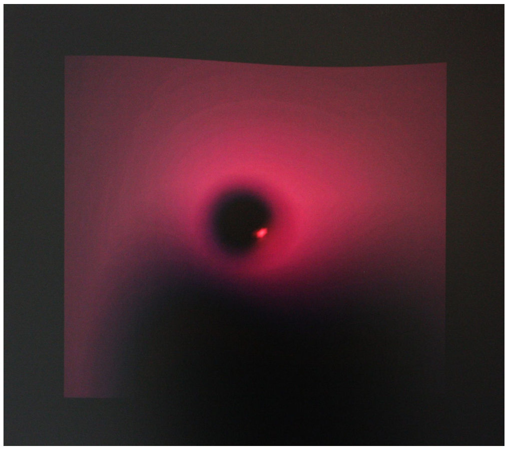


**Figure 7: Simulated light (still from a real-time generative video), (P12).** Somputueur_40 © 2014 Anne-Sarah Le Meur.

We see in these observations a number of parallels with the Technical Reasoning Hypothesis [71]. The Technical Reasoning Hypothesis posits that we have a conceptual model of technical laws or principles to devise interactions with physical objects, which conditions human's ability to detect affordances. Here the objects are not physical, but the process is similar: The technical knowledge displayed by participants is not based on physical principles that apply to physical objects, but computational principles that apply to data abstractions — or as **P1** puts it: *"It's like entering a mathematical world, it becomes abstract, and you can see invisible things"*. Such technical knowledge of the digital world is what Renom et al. call Interaction Knowledge [77, 78].

As further evidence that technical reasoning abilities influence the detection of affordances, we often observed that properties of interests of the objects that participants engage with are not manifested in directly manipulable aspects of their representation. Using Don Norman's terminology [66, 67], one might say that the affordances that matter to participants often do not have signifiers, i.e. sensory cues that make a given possibility of action salient. An example can be found with **P7** (Fig. 6) as part of a project to create 3D-printed masks: *"So rather than sculpting with clay, we were wondering what it would look like to sculpt with a fluid simulation"*.

**P12** provides similar reasoning to explain the motivation behind one of her pieces (Fig. 7): *"My initial choice is really to be working with code [...] with the concepts of light and darkness, but in a way that goes back to degree zero of computer-generated image: What can one do if you strip away as much as you can from the 3D space? What remains if it's just light - and possibly a single surface (because obviously you need a surface to intercept the rays and make them visible)?"*.

Here, the properties of interest, i.e. the properties that ought to be modified do not arise from *what* is represented by the data, but from *how* it is represented. The fact that light or a fluid become an object of interest whose properties ought to be changed is made possible by the fact that they are simulated: *"We try running a simulation of the fluid, we divide the output into a lattice, and we experiment with the type of divisions we use for this: what type of remeshing we do on the geometry, what kind of visuals we can get..."* (**P7**).

## 4.2 Co-emergence of instrumentality & materiality

To perform epistemic actions, users need to have a pre-existing mental model that enables them to perceive the materiality of the medium. But epistemic actions also contribute to enriching their mental models of materiality by uncovering affordances [32] in the materials available, or what Beaudouin-Lafon calls a *substrate*, "a digital computational medium that holds digital information" [5, 6]. Affordances of the substrate are revealed by exploring the different ways in which they can be probed and transformed into something visualizable.

For instance, point clouds are commonly used by the participants. Whether randomly generated (**P6**), imported from a model (**P1** and **P7**) or extracted from a dataset, they often form the *base material* of an exploration process: *"So for a lot of my pieces I usually start with this one node, which is called random vector. Just gives you a bunch of randomly chosen points"* (**P6**).

As a consequence, the process whereby artists uncover the creative possibilities that a given representation affords must be mediated by instruments in the sense of Beaudouin-Lafon [4]. Instruments can be viewed both as *tools-at-hand* which, by virtue of their binding with an underlying object of interest, make some of its affordances more salient, and as *tools-in-hand*, i.e. embodied effectful objects that are activated by user inputs such as mouse movements and key presses. Figure 8 illustrates the concept of instrument, using the example of the *extrude* operation in a CAD environment (in this case, OnShape [40]).

Even though the example used to illustrate this notion involves direct manipulation, non directly manipulable object can also play the role of an instrument. For instance, **P11** uses scripts as instruments: *There is a lot of geometric data that aren't represented in*

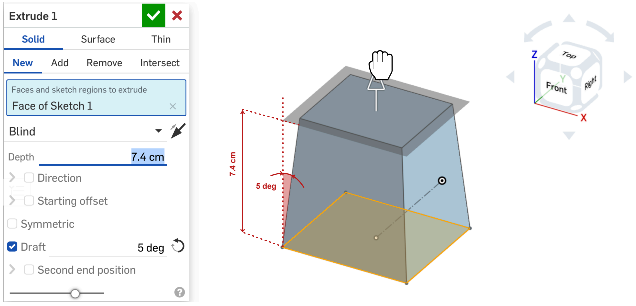


**Figure 8: Extrusion in a CAD environment (OnShape). When the *extrude* instrument is active, the profiles that afford extrusion will be highlighted in orange upon hovering and the instrument behaves as a *tool in hand*: it can be applied to any such profile in the viewport by "pulling" on it. The depth of the extrusion is mapped to cursor (and therefore hand) movements while control over the other degrees of freedom (namely direction and draft angle) is mediated by sliders.**

*the viewport. For instance Grasshopper displays the lines [generated by the procedural description of the model], but not the direction of the vectors used to generate these lines. I use scripts to visualize the missing data, it's kind of like debugging."*

As we have just outlined, detecting the affordances of a substrate — that is, making them salient — implies the use of an instrument. In order to account for how participants appropriate instruments, we need to further elaborate on the concept. An instrument is a type of *substrate* in the sense that it participates in the interaction medium, but has the added quality that its properties of interest remain unaffected by the interaction which it mediates (otherwise it could not be reused in the same way). Therefore, gaining an embodied knowledge of an instrument, or more generally, identifying what is instrumental about an object, is understanding the invariants of that object when it mediates a process of change. When appropriating instruments, participants rely on a frame of reference, i.e. they identify invariants by probing the variability allowed by the degrees of freedom of an object: *"At some point you know which parameters are relevant. At this point it's not really exploration anymore"* (**P9**).

For instance, in the visual programming language used by **P6** for procedural modeling, some operators accept inputs that are not numerical values but functions, resulting in a higher-level type of degree of freedom: *"At that point I wanted to explore the hell of this one mathematical paradigm. So for instance one thing you can do in Sverchok is vector math, and the vector math node has some interesting things but I didn't quite know what they were, so I decided to study this"* (**P2**, Fig. 9). Vector operators are fairly standard in creative programming environments, and even though documentation is available online, it is interesting to note that skilled users would rather build knowledge about them through experiments.

The idea of exploring the degrees of freedom of a system as a means to appropriate it as an instrument is also observed with **P1**: *"You always have these kinds of magic numbers, numbers that just need to be changed. So what I do is that I just change these values [Tweaks parameter] Oh! I just changed that and it changed the building size, so it must be connected to the size... and so on. If I change this it's also affecting the size, but only the height; so it's really through experimenting, trying, testing, that you discover how things work"*

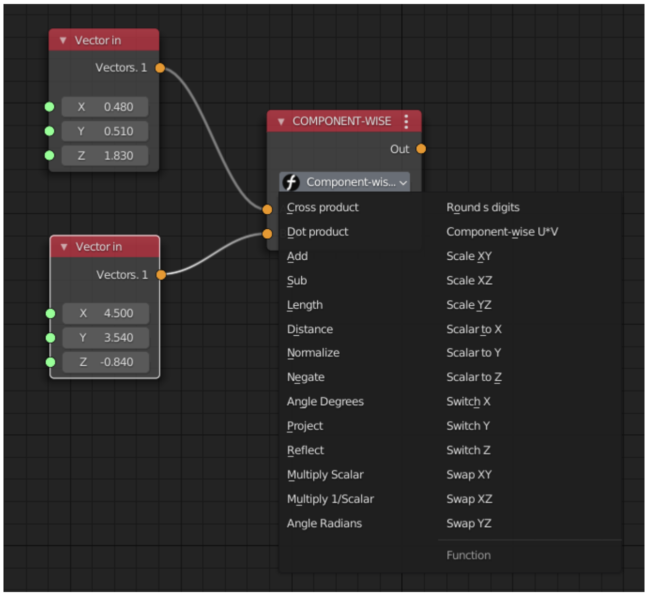
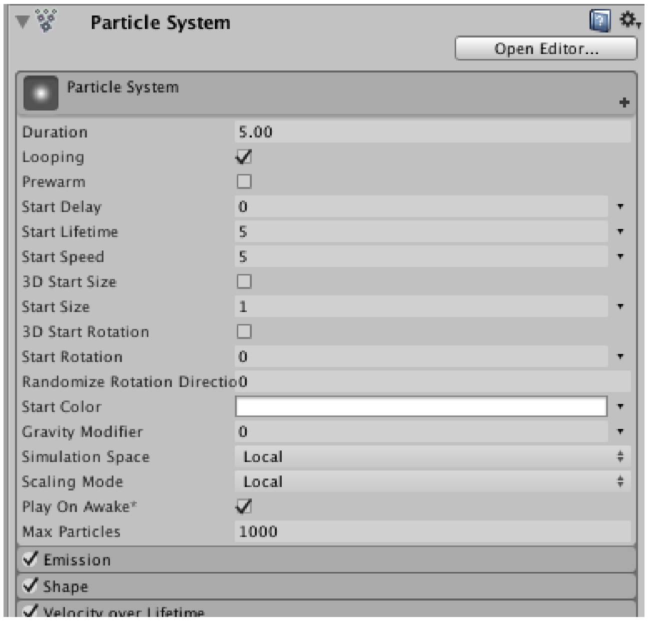


**Figure 9: Vector operator in Sverchok / Blender (P2)**

Whether these instruments are built-in features of the software or are created by the artists themselves, they frequently offer many more degrees of freedom than what a human can appropriate — often in the dozens (**P1, P2, P3, P4, P7, P9**): *"I wanted to retrieve the presets that I defined when I found exactly what I wanted. [. . . ] So for this one I had seventy (presets)"* (**P2**). *"It's endless basically. You really have lots of things. You add things, you test. You have the emission rate, so how many particles are generated, the emission shape — is it a cone, is it something else, velocity over lifetime... So you have a phenomenal amount of parameters. And it's all trial and error, you test things, you're clueless — it's crazy"*(**P9**, Fig. 10).

## 4.3 Sandbox experiments

Exploring the possibilities of an instrument — or rather of a technique that, once mastered, can be perceived by the user as an instrument — often involves the installation of a *sandbox environment*, which is explicitly devoted to epistemic actions. For example **P7**, who built a block program in Houdini in order to get an idea of the phenomena of caustics and of the way in which it could be added to his toolbox of artistic creation, explains: *"Basically this is just a very simplified ray tracer, and what I've got here is an emitter object sending rays outwards as if they were light, and they're diffracting through this object. And because it's so simplified, I'm using it as a tool to design and think about optics"*. **P8** exhibits a similar knowledge-building strategy: *"In the codebase community that I belong to, there's this concept of experiment - single purpose mini-programs that just do one thing [...] There's no goal, it's just the simplest minimum implementation of an idea, and then this becomes*



| | |
|---|---|
| Duration | 5.00 |
| Looping | ☑ |
| Prewarm | ☐ |
| Start Delay | 0 |
| Start Lifetime | 5 |
| Start Speed | 5 |
| 3D Start Size | ☐ |
| Start Size | 1 |
| 3D Start Rotation | ☐ |
| Start Rotation | 0 |
| Randomize Rotation Directio | 0 |
| Start Color | |
| Gravity Modifier | 0 |
| Simulation Space | Local |
| Scaling Mode | Local |
| Play On Awake* | ☑ |
| Max Particles | 1000 |

**Figure 10: Particle engine in Unity (P9)**

*inspiration for bigger projects, or parts of bigger projects... because now I know how it works”.*

In summary, we found evidence that participants consistently and explicitly explore the digital material (or substrates) made available through the software systems they use in order to understand its properties and capabilities. They use epistemic actions, in particular through both existing and created instruments, to gain this knowledge, through an exploratory process.

## 5 EPISTEMIC ARTIFACTS

Early on in the interview process, a quote by **P1** caught our attention: *“When you’re doing creative coding you’re always somewhere between making art and making tools”.* Many other observations were consistent with that statement, but at first, this perspective seemed completely orthogonal to the question of epistemic action. A quote from Simondon turned out to be the key that unlocked a new angle of analysis: *“What resides in the machines is the human reality, the human gesture fixed and crystallized in structures that work”*[1] [88]. Envisioning artifacts as solidified knowledge prompted us to revisit many untagged units of meaning in our data through the lens of what we call *epistemic artifacts.* As a result, many observations that would have otherwise appeared anecdotal took on a new dimension.

### 5.1 Proofs of virtuosity

For instance, we noticed that producing design variants was commonplace, even for projects that had no design brief, client requirements nor performance constraints that would otherwise mandate producing different solutions. At a time when autonomous systems that are able to generate complex images with the simple click of a mouse are becoming mainstream, artists and designers need to prove that their work is not just the result of a lucky accident, nor of blindly executing an algorithmic recipe, but that there is an actual, unique underlying know-how. Under such a perspective, one can indeed think of artworks as being themselves “solidified knowledge”. Making “series” of artworks (e.g. Fig. 11) is a way to showcase the time, dedication and expertise that goes into understanding the design space of a computational tool: *“What takes the most time is making choices. Choosing which frame to pick, choosing to stop here or to carry on”* (**P9**).

Whether the outputs an artist produces are the culmination of a search for novelty, or solutions to constraints expressed in a brief, their *novelty* and *value* matter little if they do not incorporate any of the user’s personal know-how: *“There was a point where I started to experiment with this reaction-diffusion algorithm. That’s much more what I was actually aiming at. And there’s actually a plugin for that, a Blender plugin [...] But I find it really boring in the end. [...] It’s a solved problem: reaction-diffusion, topology optimization, that kind of look is a solved problem. [...] I’m quite glad I didn’t end up going that way. Because just pressing the simulate button in Blender was not satisfying, it didn’t give me this ’I made this’ feeling.”* (**P6**, Fig. 2).

These “proofs of virtuosity” might be stills, videos or even interactive pieces that blur the boundary between an art piece (or a design) and a tool. For instance, the realizations on display on **P8**’s portfolio showcase both the visual result and the interface that was designed to control it: *“We’re not shy about the interface. We’re kind of proud of the fact that this is software and that there are sliders and buttons and presets”* (Fig. 12).

The epistemic nature of P10’s artistic productions was perhaps the most explicit, since the core motivation for many of his projects is to produce works that highlight the *computational* nature of the objects of interest he manipulates. Fig. 13 is a series of 3D resin engravings in which 3d models are deconstructed into sequences

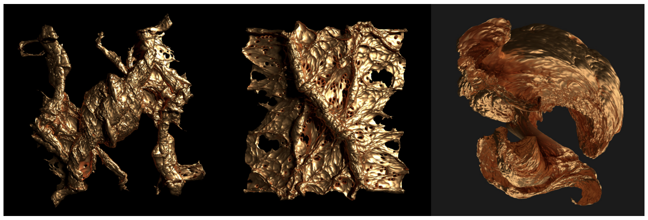


**Figure 11: Experiments on the procedural deformation of 3D meshes (P6).** *Slices of the ocean.* © 2017 Jakub Fiala.

---

[1] Original quote: “Ce qui réside dans les machines, c’est la réalité humaine, du geste humain fixé et cristallisé en structures qui fonctionnent”

of lines that correspond to the order in which various file formats encode geometric primitives.

## 5.2 Traces of exploration

Traces of exploration are another example of knowledge externalized into persistent representations. We define them as snapshots that document successive steps of the process and capture knowledge that might need to be reused later on, by the artist or by someone else. Traces of exploration may include interesting findings that are good candidates to be showcased as proofs of virtuosity later on: *"Well the idea essentially is to work like a photographer, trying to capture an evolving object. I set up a system and whenever it reaches a state that I like I capture it, and I have a whole series of variations that I've saved"* (**P4**, Fig. 4).

The core difference between traces of exploration and proofs of virtuosity is that the information that is externalized in the former is meant to be more unambiguously reusable than the latter: If a chef's dish were a proof of virtuosity, traces of explorations would be better framed as the ingredients or steps of their recipes. While a static artwork can be reused, at best, as a source of inspiration or as "raw" material for future work, the screenshots captured by **P4** also include the slider interface so as to recall the corresponding input configuration. In a similar fashion, **P1** captures his whole workspace through screenshots, while **P6** includes the value of the random seed from which he derives variations in the design space in the name of the saved file.

Despite efforts to document their process, navigating the history of a project is not always easy, because the logic behind a sequence of actions is not necessarily linear and cannot be meaningfully retrieved using undo/redo actions: *"let me just reopen this because I must have been unplugging things a bit too wildly"* (**P6**). The difficulty to navigate back into the history of a document can also be attributed to the context-dependent nature of instruments and objects of interest. Both **P9** and **P4** reported struggling with reopening previous workspaces because updates in the environment and its dependencies were breaking. For several participants (**P4**, **P8**, **P9**), traces of exploration, especially in the early phase of a project, were better captured with pictures of brainstorming dashboards or handwritten notes: *"So first we wrote a storyboard of things we wanted to have, on paper"* (**P9**).

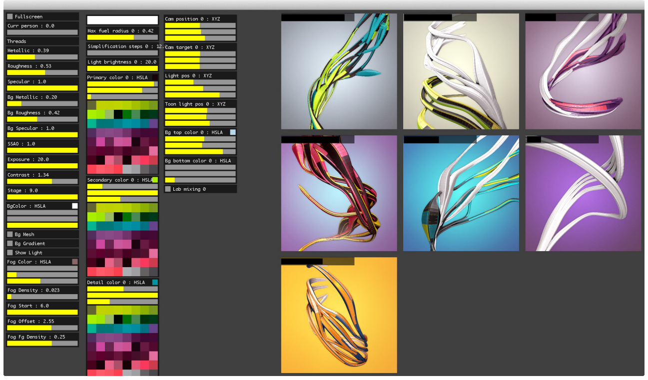

Fullscreen
Curr person : 0.0
Threads
Metallic : 0.39
Roughness : 0.53
Specular : 1.0
Bg Metallic : 0.20
Bg Roughness : 0.42
Bg Specular : 1.0
SSAO : 1.0
Exposure : 20.0
Contrast : 1.34
Stage : 9.0
BgColor : HSLA
Bg Mesh
Bg Gradient
Show Light
Fog Color : HSLA
Fog Density : 0.023
Fog Start : 6.0
Fog Offset : 2.55
Fog Fg Density : 0.25
Max fuel radius 0 : 0.42
Simplification steps 0 : 12.
Light brightness 0 : 20.0
Primary color 0 : HSLA
Secondary color 0 : HSLA
Detail color 0 : HSLA
Cam position 0 : XYZ
Cam target 0 : XYZ
Light pos 0 : XYZ
Toon light pos 0 : XYZ
Bg top color 0 : HSLA
Bg bottom color 0 : HSLA
Lab mixing 0

**Figure 12: Bespoke interface developed by P8 to generate artistic data-visualizations for the project *Fibers* [45]**

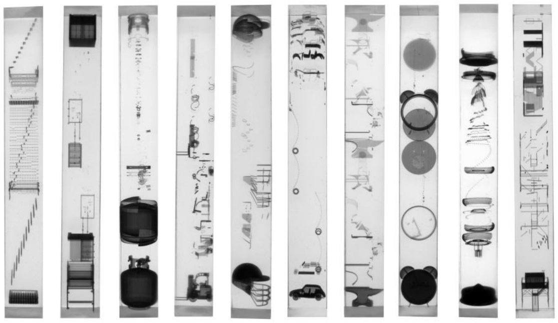


**Figure 13: Exploded view of 3D models, 3D-engraved in resin (P10).** *Clickspace II.* © 2015 Jonah Marrs **[59]**

## 5.3 Bespoke instruments

Some of the most expert participants routinely build their own tools: *"We build such tools for ourselves pretty much every time"* (**P8**). Unlike traces of exploration, such bespoke instruments do not capture a state, but a space of possibilities that has been created and optimized with the explicit goal of being reused. Building bespoke instruments can be viewed as a consequence of the importance of epistemic actions, since instruments mediate these actions.

For example, once an artist has found a parameterized procedure that maps to an interesting design space and has identified possible outliers in that design space as well as how two input parameters might interfere with each other, it often makes sense to try and embed this informal knowledge into an interface that implements a more opinionated way of interacting with that parameterized procedure. Input remapping is a good example of such a process: *"a big chunk of my work goes into remapping intervals and ranges that I figured would be relevant based on what I'm building"* (**P3**); *"Because these are systems that, again, are data driven, or that need to be constrained in terms of the output they are producing, we normalize whatever the values are from 0 to 1, to have an understandable mapping between input and output and boundaries"* (**P8**).

Building bespoke tools, however, comes at a cost. Unlike saving snapshots of exploration, which is essentially a "free" operation in terms of time and cognitive load, building bespoke tools is time-consuming and is only warranted if one is certain that it will be reused. For users such as **P7**, it only makes sense if opportunities for reuse have clearly been identified: *"You could always remap the linear sliders to something else, but I think so often it's not really worth it to build that. [...] there's always a kind of, kind of a balance that we've struck between automating things and doing them manually... and it only pays off if you use it more than once".*

The relevance of capturing this knowledge about the behavior of tools becomes even more striking when considering that such epistemic artifacts are often shared, e.g. with a client (**P8**, Fig. 14) or

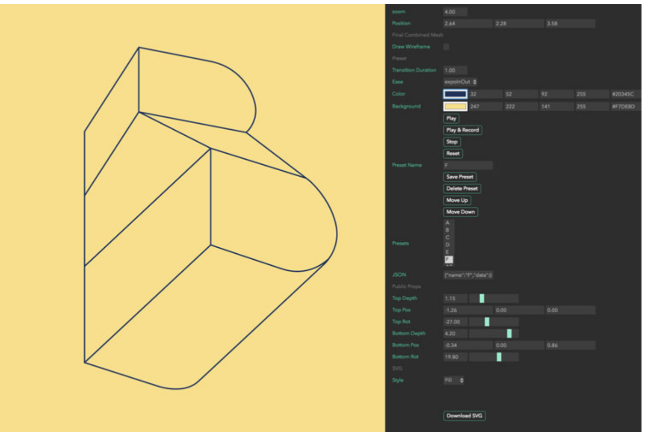

zoom 4.00
Position 2.64 2.28 3.58
Final Combined Mesh
Draw Wireframe ☐
Preset
Transition Duration 1.00
Ease expoInOut
Color 32 52 92 255 #20345C
Background 247 222 141 255 #F7DE8D
Preset Name F
Presets A B C D E F
JSON {"name":"F","data":{}
Public Props
Top Depth 1.15
Top Pos -1.26 0.00 0.00
Top Rot -27.00
Bottom Depth 4.20
Bottom Pos -0.34 0.00 0.86
Bottom Rot 19.80
SVG
Style Fill

**Figure 14: Bespoke interface designed by P8 to let a client customize a generative logo. [46]**

with people whose involvement in the creative process is closer to that of a curator or an art director: *"Then there are situations where you're going to want to expose some parameters to someone who's not here to actively contribute and get into the details, but who's just here to give feedback and get to the essence of it, get a broad overview of the different versions, in such cases you try to expose as few parameters as possible"* (**P3**). Symmetrically, **P11** explains that the curator (in his case the lead architect) sometimes wants to edit inputs that have been defined as constants by the computational designer, which implies that the design space currently accessible is too limited with regards to the lead architect's expectations and needs to be expanded by adding degrees of freedom.

Instruments are sometimes created for situations that mandate an even more extreme form of reuse than possible with bespoke tools, namely live interaction. In some cases, participants anticipate live interaction with a procedural model to be one of the ultimate goals: *"At the beginning I really thought it up as a musical instrument, even though I regret it a bit today"* (**P4**), *"With this tool, we wanted to be able to improvise completely"* (**P3**). In other cases, the process was more serendipitous: **P1**, **P4**, **P9** and **P12** reported that interacting with the sliders had made them realize that temporal trajectories within the design space carried intrinsic artistic values, which prompted them to repurpose the corresponding parameterized procedures into a playable visual instrument that they use during live performances.

In this type of scenario, a frequent constraint is that the number of inputs must remain manageable: *"Here for instance I'm working on an object called 'Parallel Lives', that has so many available options that when I'll integrate it into [P4's custom built VJing environment] I'll have to remove three quarters of the parameters"* (Fig. 4) **P4**). Moreover, their mapping to the design space must be relevant from the perspective of gestural expressivity: *"So when you're building visual effects you have to keep in mind what the dependencies between parameters are, so that the higher-level parameters that you expose aren't interdependent"* (**P3**). This finding is consistent with previous work on digital musical instruments [43]. As a matter of fact, our data contained multiple situated observations that are consistent with research results on musical interfaces. We address this point further in the next section.

In summary, through the lens of epistemic artifacts, we observed that *final* artworks or designs, traces of exploration and self-made tools capture, each in their own way, the intimate knowledge that participants have of their medium. As they form new substrates and mediate new interactions, epistemic artifacts also contribute to enriching the medium itself with new possibilities of exploration.

## 6 DISCUSSION

After discussing the issue of gender imbalance in our study, we address the implications of this work for the development of tools that better support the epistemic process that we have illustrated, and towards an evaluation framework based on these concepts.

### 6.1 Gender diversity

The gender diversity of our study participants was wildly imbalanced (11 men, 1 woman), which raises the question of whether this is representative of the population. Data regarding gender representation in the fields targeted by our study, which can be characterized as lying at the intersection of visual arts and programming, is very scarce. A 2021 report by the USC Annenberg Inclusion Initiative [47] shows a large imbalance in the field of Visual Effects (VFX), with 21.6% of women in title credits and even lower percentages in leadership roles. Representation of women is only slightly better documented in the adjacent field of Music Technology. Born & Devine's study on gender in computer music reports that Music Technology students in the UK were 90 percent male [12]. Anecdotal examples in a study by Mori [64] further highlight the under-representation of women in the field of live coding. These trends are reflected in the demographics of our study, and future work should adopt sampling strategies that actively counterbalance the under-representation of women in creative computing.

A deeper issue however is whether our findings can inform the design of systems that do not perpetuate the dynamics that keep women a minority in our field of interest. The early history of computer art makes a compelling case that neither skills in programming, visual arts and designs, nor the ability to combine the two are gender-related traits [98]. Women played a pioneering role in exploring and demonstrating the potentialities of computing as a creative medium, with Lilian Schwarz and Vera Molnar often cited as the precursors of digital art.

A reason often advanced to explain why STEM-related activities — specifically computing — have come to be dominated by men is that women tend to become marginalized in a given field as soon as being active in that field is perceived as yielding power or prestige [104]. In accordance with that view, we posit that the off-the shelf definition of creativity that prevails in computing, which labels behaviour as "creative" by virtue of what is essentially a "fitness criterion" (i.e. novelty and value), perpetuates this type of power dynamics. Our hope is that examining creative behaviours without this preconceived view of creativity can lead to design principles that are ultimately more inclusive.

## 6.2 Input techniques to support epistemic actions

From a systems-building perspective, a number of design opportunities emerge from our work. One of the most salient issues encountered by artists and designers is "slider tweaking", which had already been identified 25 years ago as one of the great vexations of computer graphics [58]. More generally, we observe that adjusting a configuration of parameters is particularly tedious: *"Choosing [visual, e.g. Perlin] noise can be very long, and some of my nights are spent tweaking these noise values and how it looks like"* (**P1**). Whether it involves controlling the position of a cursor, selecting an option in a drop-down menu or toggling a checkbox, the pointer tool is ill-suited to support both coarse-grained exploratory tasks on large input spaces and fine-grained modulation tasks. This situation motivates a number of workaround and strategies observed among participants, including (1) mapping input parameters to a MIDI controller (**P1**, **P2**, **P3**, **P4**, **P9**) or to keyboard shortcuts (**P12**), and (2) defining macro-mappings (**P3**, **P11**).

The use of Musical Instrument Digital Interface (MIDI) controllers reinforces our belief that one of the first bottlenecks to epistemic actions is the fact that conventional input devices (mouse, keyboard, touchscreen) do not take full advantage of the number of degrees of freedom of the human sensorimotor system, especially the hand, nor of its expressive range. This often results in convoluted user interfaces that rely on modes and hardly discoverable shortcuts.

The field of music interfaces shows a way forward, with the adoption of an interoperability standard for controlling interfaces with physical peripherals. Indeed, the MIDI protocol enables multiple parallel inputs on physical interfaces where haptic feedback can replace the visual feedback of a graphical user interface (GUI), thereby letting the user concentrate on the visual output of the process being controlled. While MIDI controllers still rely mostly on knobs, faders and buttons, the recently adopted MIDI Polyphonic Expression standard (MPE) [2] allows for sophisticated attack-decay-sustain-release responses, expanding the expressive power of MIDI controllers. Even a simple addition to the mouse and trackpad to sense pressure and after-touch would dramatically increase the epistemic capabilities of standard input devices.

Low-level software aspects also contribute to this bottleneck. The lack of support for composite input events considerably limits the range of both pragmatic and epistemic actions one can perform. For example, simultaneous touch and stylus input are not handled on the Apple iPad tablet; major operating systems do not support multiple pointers; and making UI actions accessible through a transparent API is rarely a concern of front-end developers. Yet it is the ability to combine inputs and input techniques that allows users to learn and exploit complex gesture vocabularies, as demonstrated by previous work, e.g. on bimanual interaction [9, 20], pen+touch interaction [39] and chord-based interaction [31].

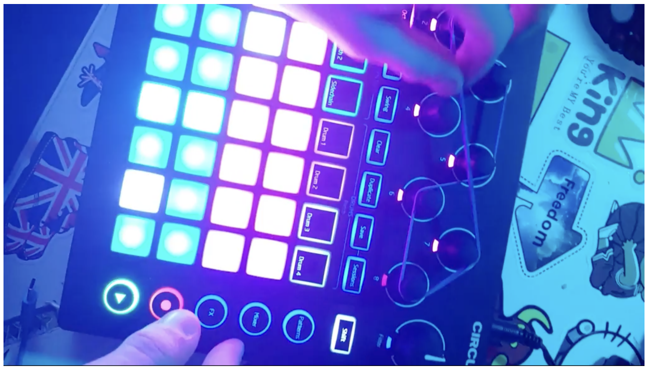


**Figure 15: Workaround from an anonymous Reddit user to control multiple knobs at once**

## 6.3 Supporting the creation of epistemic artifacts

Defining macro-mappings is representative of the fact that it is inherently impossible for designers to anticipate which instruments should be provided to the end user, since the ability to perceive and compose new objects of interest often leads participants to have their own idea of how such objects ought to be interacted with: *"The interaction comes from playing with the parameters. Whenever you feel like one parameter is interesting to change, you assign it to an element on the controller"* (**P1**).

However, what is tedious for users is that they need to carefully craft the transfer function that maps the input signal, such as the position of the mouse over time, to the modification of several objects of interest, such as simultaneously moving several sliders. Such tasks typically involve a high cognitive load, because they require translating the user's intention into a mathematical formula. As a consequence, some participants feel that the cost of building a bespoke tool outweighs its benefits unless a simple practical solution is found. For example, an anonymous Reddit user has posted a video [101] of a setup where he uses rubber bands to maintain mechanical constraints between different knobs (Fig. 15). For a more general approach, we suggest that interfaces that support declaring and maintaining bidirectional constraints between objects of interest could be a solution.

The example in Fig. 15 shows how declarative constraints can be viewed as a way of embedding knowledge about how the different objects of interest in a substrate should influence each other. This principle has been used for a long time in Computer-Aided Design (CAD) environments. It was featured in SketchPad, the very first GUI [95] and was advocated as a paradigm for software architectures, in particular by Borning [13, 14]. Unfortunately, constraint-based architectures for interaction have not made it to mainstream software, depriving end-users of sophisticated customization capabilities.

## 6.4 Towards an evaluation framework

We are also interested in operationalizing the concepts of epistemic actions and artifacts into an evaluation framework. This task has been partially initiated by Fjeld & Barendregt [29], who studied the use of epistemic actions as a metric for evaluating interfaces. Their findings indicate that epistemic action is a measure that is independent of the three traditional usability measures, i.e. efficiency, effectiveness, and satisfaction.

At present, the Creativity Scoring Index [25] is one of the most frequently used frameworks for evaluating creativity support in interfaces, but none of its dimensions capture the epistemic aspects of creativity that we have identified in this article. In future work, we intend to revisit the Creativity Scoring Index questionnaire based on these findings, using qualitative assessment of both the support for epistemic action and its counterpart, i.e. the externalization of knowledge into epistemic artifacts. For example, the following Likert items could be used to assess the support for epistemic actions:

- When engaging with the interface, the results of my actions gave me ideas for other actions I could or should try to undertake;
- I was able to successfully infer from my previous knowledge how to execute the actions that came to my mind;
- My ability to anticipate the effect of my actions improved over the course of the task.

In the field of visualization, Sedlmair et al. [84] propose a taxonomy of exploratory actions frequently observed in scientific visualization and simulation workflows. These workflows bear many similarities with the workflows of computer-generated effects software, and the classes they identify overlap with the themes covered by the sample Likert items above.

While the work of Sedlmair et al. does not lead to an evaluation framework, this task has been initiated by Stasko [93], who proposes a questionnaire-based method to assess the amount of valuable information acquired by a user from its engagement with an interactive visualization. However, the proposed approach is constrained by the assumption that *knowledge* and *meaning* preexist in the data or the model with which one interacts, and is simply waiting to be discovered using the right visualization. Groth et al. [34] had previously recognized that this is not always the case, and identified the unaddressed need of users to complement their visualisations with annotations that embed knowledge external to the task domain. Our observations push this point further as they indicate that expert users produce new — and often tacit — knowledge and meaning in ways that cannot always be captured by simple annotations.

Designers create epistemic artifacts precisely for the purpose of capturing, reusing and sharing this knowledge. The ability to create such artifacts depends on features such as naming objects and collections of parameter settings, saving and recalling them, structuring them into hierarchies or networks, creating and executing scripts, customizing the mapping of gestures and device-inputs to behaviours of the interface, etc. A questionnaire could therefore assess the availability and effective use of these features, and more precise measurements could be performed by asking users to perform specific tasks that require externalisation of their acquired knowledge.

## 7 CONCLUSION

This paper investigates creative practices where computing is the primary medium for producing visual output, e.g. through the use of creative coding or generative design. These practices entail a deep understanding of the mediating properties of code, data and computational representations for producing novel and valuable artifacts. We conducted an observational study with 12 artists and designers who create and/or configure algorithmic procedures to produce visual works. The first contribution of this work is therefore of an empirical nature: our observations supplement the existing body of research work on emerging computational practices in visual creation, emphasizing the various techniques used by artists and designers to explore and probe the capabilities of their tools.

Our analysis is driven by a theoretical framework based on the notion of epistemic action [52] and drawing from the concepts of materiality [15], affordances [32] and technical reasoning [71]. The second contribution of this work is therefore of a theoretical nature: we reframe creative behavior as an epistemic process, whereby authors generate knowledge about their medium via *epistemic actions* and externalize this new knowledge into what we call *epistemic artifacts*. Epistemic artifacts are created explicitly for the sake of exploration and understanding rather than for progressing toward the creative goal.

This work contributes a deeper understanding of the roles of instrumentality and materiality in the creative process. It opens up new perspectives to better support this epistemic process in software and to define evaluation frameworks that account for it. Future work should expand the scope of this work to other areas of creativity-support tools, e.g. in the sound and music area, as well as to a more diverse set of participants. The concepts of epistemic actions and artifacts could also be applied to other areas than creativity, such as in visualisation when exploring an ill-defined problem space. In the long run, we seek to identify concepts and principles for a generative theory of interaction [7] that captures the epistemic aspects of computer-based creative processes and enables the emergence of computation as a rich, creative medium.

## ACKNOWLEDGMENTS

We would like to thank the artists who made this research possible by sharing their time and their work: *Léon Denise, Élie Michel, Jonathan Giroux, Stanislas Marçais, Jakub Fiala, Duncan Carter, Mick Geerits, Marcin Ignac, Fabrice Starzinskas, Jonah Marrs, Wael Ben Mansour, Anne-Sarah Le Meur*

This project was partially supported by a grant from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 695464 ONE).

# REFERENCES

[1] Mattias Arvola and Johan Linder. 2018. Know thy users by interpretative phenomenological analysis. *Journal of Interaction Science* 6, 3 (2018). https://doi.org/10.24982/jois.1719018.003

[2] MIDI Manufacturers Association. 2018. *MIDI Polyphonic Expression*. MIDI Manufacturers Association, Los Angeles, CA, United States. Version 1.0.

[3] Janet Banfield and Mark Burgess. 2013. A phenomenology of artistic doing: Flow as embodied knowing in 2D and 3D professional artists. *Journal of Phenomenological Psychology* 44, 1 (2013), 60–91.

[4] Michel Beaudouin-Lafon. 2000. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. In *Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, United States, 446–453. https://doi.org/10.1145/332040.332473

[5] Michel Beaudouin-Lafon. 2017. Towards Unified Principles of Interaction. In *Proceedings of the 12th Biannual Conference of the Italian SIGCHI Chapter* (Cagliari, Italy) *(CHItaly '17)*. ACM, New York, NY, United States, Article 1, 2 pages. https://doi.org/10.1145/3125571.3125602

[6] Michel Beaudouin-Lafon. 2023. Beyond Applications: Interaction Substrates and Instruments. In *Proceedings of the 34th Conference de l'association francophone de l'Interaction Humain-Machine* (TROYES, France) *(IHM '23)*. Association for Computing Machinery, New York, NY, United States, Article 7, 15 pages. https://doi.org/10.1145/3583961.3583968

[7] Michel Beaudouin-Lafon, Susanne Bødker, and Wendy E. Mackay. 2021. Generative Theories of Interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)* 28, 6, Article 45 (November 2021), 54 pages. https://doi.org/10.1145/3468505

[8] Ilias Bergstrom and R Beau Lotto. 2015. Code Bending: A new creative coding practice. *Leonardo* 48, 1 (2015), 25–31.

[9] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. 1993. Toolglass and Magic Lenses: The See-through Interface. In *Proc. 20th Annual Conference on Computer Graphics and Interactive Techniques* (Anaheim, CA) *(SIGGRAPH '93)*. ACM, New York, NY, United States, 73–80. https://doi.org/10.1145/166117.166126

[10] Margaret A Boden. 2009. Computer models of creativity. *AI Magazine* 30, 3 (2009), 23–23.

[11] Margaret A Boden and Ernest A Edmonds. 2009. What is generative art? *Digital Creativity* 20, 1-2 (2009), 21–46.

[12] Georgina Born and Kyle Devine. 2016. Gender, creativity and education in digital musics and sound art. *Contemporary Music Review* 35, 1 (2016), 1–20.

[13] Alan Borning. 1981. The Programming Language Aspects of ThingLab, a Constraint-Oriented Simulation Laboratory. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 3, 4 (October 1981), 353–387. https://doi.org/10.1145/357146.357147

[14] Alan Borning and Robert Duisberg. 1986. Constraint-Based Tools for Building User Interfaces. *ACM Transactions on Graphics (TOG)* 5, 4 (October 1986), 345–374. https://doi.org/10.1145/27623.29354

[15] Tone Bratteteig. 2010. *A Matter of Digital Materiality*. Springer London, London, United Kingdom, 147–169. https://doi.org/10.1007/978-1-84996-223-0_5

[16] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

[17] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[18] Andrew Brown. 2012. Creative Partnerships with Technology: How creativity is enhanced through interactions with generative computational systems. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 8, 4 (2012), 14–20. Issue AIIDE Workshop Technical Report WS-12-16. https://doi.org/10.1609/aiide.v8i4.12555

[19] Mark Burry. 2011. *Scripting cultures: Architectural design and programming*. John Wiley & Sons, Chichester, England.

[20] W. Buxton and B. Myers. 1986. A Study in Two-Handed Input. In *Proc. SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, United States) *(CHI '86)*. ACM, New York, NY, United States, 321–326. https://doi.org/10.1145/22627.22390

[21] Manuel Bärenz. 2020. The Essence of Live Coding: Change the Program, Keep the State!. In *Proceedings of the 7th International Workshop on Reactive and Event-Based Languages and Systems* (Chicago, MI, United States) *(REBLS '20)*. ACM, New York, NY, United States, 2–14. https://doi.org/10.1145/3427763.3428312

[22] Claude Cadoz. 1988. Instrumental gesture and musical composition. In *Proc. 14th International Computer Music Conference* (Cologne, Germany) *(ICMC '88)*. Michigan Publishing, Ann Arbor, MI, United States, 1–12.

[23] Claude Cadoz, Annie Luciani, Jérôme Villeneuve, Alexandros Kontogeorgakopoulos, and Iannis Zannos. 2014. Tangibility, presence, materiality, reality in artistic creation with digital technology. In *Proc. joint 40th International Computer Music Conference and 11th Sound & Music Computing Conference (ICMC/SMC '14)*. Michigan Publishing, Ann Arbor, MI, United States, 754–761.

[24] Duncan Carter and Mick Geerits. 2018. Algorithms Unmasked. http://abnormal.design/algorithmsunmasked

[25] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction* 21, 4, Article 21 (2014), 25 pages. https://doi.org/10.1145/2617588

[26] Simon Colton and Geraint A. Wiggins. 2012. Computational creativity: The final frontier?. In *Proc. European Conference on Artificial Intelligence* (Montpellier, France) *(ECAI '12, Vol. 12)*. European Association for Artificial Intelligence, IOS Press, Amsterdam, Netherlands, 21–26.

[27] Alan Dix. 2007. Designing for Appropriation. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers, Vol. 2* (Lancaster, United Kingdom) *(BCS-HCI '07)*. BCS Learning & Development Ltd., Swindon, United Kingdom, 27–30.

[28] Alan Dorin. 2013. Aesthetic selection and the stochastic basis of art, design and interactive evolutionary computation. In *Proceedings of the Conference on Genetic and Evolutionary Computation* (Amsterdam, The Netherlands) *(GECCO '13)*. ACM, New York, NY, United States, 311–318.

[29] Morten Fjeld and Wolmet Barendregt. 2009. Epistemic action: A measure for cognitive support in Tangible User Interfaces? *Behavior research methods* 41, 3 (2009), 876–881.

[30] Jonas Frich, Michael Mose Biskjaer, and Peter Dalsgaard. 2018. Twenty Years of Creativity Research in Human-Computer Interaction: Current State and Future Directions. In *Proceedings of 2018 ACM Conference on Designing Interactive Systems* (Hong Kong, China) *(DIS '18)*. ACM, New York, NY, United States, 1235–1257.

[31] Emilien Ghomi, Stéphane Huot, Olivier Bau, Michel Beaudouin-Lafon, and Wendy E Mackay. 2013. Arpège: learning multitouch chord gestures vocabularies. In *Proc. 2013 ACM international conference on Interactive tabletops and surfaces*. ACM, New York, NY, United States, 209–218.

[32] James Gibson. 1979. *The Ecological approach to visual perception*. Psychology Press, London, England, United Kingdom.

[33] Shad Gross, Jeffrey Bardzell, and Shaowen Bardzell. 2014. Structures, forms, and stuff: the materiality and medium of interaction. *Personal and Ubiquitous Computing* 18 (2014), 637–649.

[34] D.P. Groth and K. Streefkerk. 2006. Provenance and Annotation for Visual Exploration Systems. *IEEE Transactions on Visualization and Computer Graphics* 12, 6 (2006), 1500–1510. https://doi.org/10.1109/TVCG.2006.101

[35] Armand Hatchuel. 2001. Towards design theory and expandable rationality: the unfinished program of Herbert Simon. *Journal of Management and Governance* 5, 3/4 (2001), 260–273.

[36] Armand Hatchuel and Benoit Weil. 2003. A new approach of innovative Design: an introduction to CK theory. In *Proceedings of the 14th International Conference on Engineering Design* (Stockholm, Sweden) *(ICED '03)*. Design Society, Glasgow, Scotland, United Kingdom, Article 1794, 15 pages.

[37] Mary Hegarty. 2004. Mechanical reasoning by mental simulation. *Trends in Cognitive Sciences* 8, 6 (2004), 280–285. https://doi.org/10.1016/j.tics.2004.04.001

[38] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. 2013. Procedural Content Generation for Games: A Survey. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* 9, 1, Article 1 (February 2013), 22 pages. https://doi.org/10.1145/2422956.2422957

[39] Ken Hinckley, Koji Yatani, Michel Pahud, Nicole Coddington, Jenny Rodenhouse, Andy Wilson, Hrvoje Benko, and Bill Buxton. 2010. Pen + Touch = New Tools. In *Proceedings of the 23nd ACM symposium on User Interface Software and Technology* (New York, NY, United States) *(UIST '10)*. ACM, New York, NY, United States, 27–36. https://doi.org/10.1145/1866029.1866036

[40] John Hirschtick, John McEleney, Michael Lauer, Ravi Reddy, Paul Chastell, and Ilya Baran. 2012. OnShape.

[41] Jonathan Hook, David Green, John McCarthy, Stuart Taylor, Peter Wright, and Patrick Olivier. 2011. A VJ Centered Exploration of Expressive Interaction. In *Proc. SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. ACM, New York, NY, United States, 1265–1274. https://doi.org/10.1145/1978942.1979130

[42] Jonathan Hook, John McCarthy, Peter Wright, and Patrick Olivier. 2013. Waves: Exploring Idiographic Design for Live Performance. In *Proc. SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. ACM, New York, NY, United States, 2969–2978. https://doi.org/10.1145/2470654.2481412

[43] Andy Hunt and Marcelo M. Wanderley. 2002. Mapping Performer Parameters to Synthesis Engines. *Organised Sound* 7, 2 (August 2002), 97–108. https://doi.org/10.1017/S1355771802002030

[44] Edwin L Hutchins, James D Hollan, and Donald A Norman. 1985. Direct manipulation interfaces. *Human–computer interaction* 1, 4 (1985), 311–338.

[45] Marcin Ignac. 2014. Fibers. https://variable.io/fibers/

[46] Marcin Ignac and Nick Nikolov. 2018. Broadgate: A bespoke animation tool for a kinetic identity. https://variable.io/broadgate/

[47] USC Annenberg Inclusion Initiative. 2021. *Invisible in Visual Effects: Understanding the Prevalence and Experiences of Women in the Field*. Technical Report. USC Annenberg School for Communication and Journalism. https://assets.uscannenberg.org/docs/aii-study-women-in-visual-effects-2021-11-04.pdf

[48] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, GA, United States) *(CHI '97)*. ACM, New York, NY, United States, 234–241. https://doi.org/10.1145/258549.258715

[49] Jennifer Jacobs, Joel Brandt, Radomír Mech, and Mitchel Resnick. 2018. Extending Manual Drawing Practices with Artist-Centric Programming Tools. In *Proc. 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, United States, 1–13. https://doi.org/10.1145/3173574.3174164

[50] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction (TEI '07)*. ACM, New York, NY, United States, 139–146.

[51] David Kirsh. 2010. Thinking with external representations. *AI & Society* 25, 4 (Feb. 2010), 441–454. https://doi.org/10.1007/s00146-010-0272-8

[52] David Kirsh and Paul Maglio. 1994. On distinguishing epistemic from pragmatic action. *Cognitive Science* 18, 4 (1994), 513–549. https://doi.org/10.1016/0364-0213(94)90007-8

[53] Jingyi Li, Joel Brandt, Radomír Mech, Maneesh Agrawala, and Jennifer Jacobs. 2020. Supporting Visual Artists in Programming through Direct Inspection and Control of Program Execution. In *Proc. 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, United States) *(CHI '20)*. ACM, New York, NY, United States, 1–12. https://doi.org/10.1145/3313831.3376765

[54] Johan Linder and Mattias Arvola. 2017. IPA in UX Research: Interpretative phenomenological analysis in a user experience design practice. In *Proc. European Conference on Cognitive Ergonomics*. IEEE, New York, NY, United States, 17–24.

[55] Lambros Malafouris. 2004. *The Cognitive Basis of Material Engagement: Where Brain, Body and Culture Conflate.* Cambridge University Press, Cambridge, United Kingdom, Chapter 5, 53–61. http://cogprints.org/4629/

[56] Lambros Malafouris. 2020. Thinking as "thinging": Psychology with things. *Current Directions in Psychological Science* 29, 1 (2020), 3–8.

[57] Stanislas Marçais. 2019. La Joconde à l'Hermine. https://www.knupel.art/art/

[58] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. 1997. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In *Proc. 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 389–400. https://doi.org/10.1145/258734.258887

[59] Jonah R. Marrs. 2015. Clickspace: Visualizing Computer Processes Through Time-Lapse Representation. *Shift: Graduate Journal of Visual and Material Culture* 10 (2015), 1. http://shiftjournal.org/seam/clickspace/

[60] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. 2006. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine* 27, 4 (2006), 12–12.

[61] Alex McLean. 2014. Making programming languages to dance to: live coding with tidal. In *Proceedings of the 2nd ACM SIGPLAN International Workshop on Functional Art, Music, Modeling & Design* (Gothenburg, Sweden) *(FARM '14)*. ACM, New York, NY, United States, 63–70. https://doi.org/10.1145/2633638.2633647

[62] Alex McLean and Geraint A. Wiggins. 2010. Bricolage Programming in the Creative Arts.. In *Proc. Annual Worshop of the Psychology of Programming Interest Group*. Universidad Carlos III de Madrid, Madrid, Spain, 18.

[63] Richard Menary. 2010. Introduction to the special issue on 4E cognition. *Phenomenology and the Cognitive Sciences* 9 (2010), 459–463.

[64] Giovanni Mori. 2015. Analysing Live Coding with Ethnographical Approach: A New Perspective. In *Proceedings of the 1st International Conference on Live Coding* (Leeds, United Kingdom) *(ICLC '15')*. University of Leeds Interdisciplinary Centre for Scientific Research in Music, Leeds, United Kingdom, 13–15.

[65] Allen Newell, J. C. Shaw, and Herbert A. Simon. 1962. The processes of creative thinking. In *Proceedings of the 1958 Symposium on Contemporary Approaches to Creative Thinking* (Boulder, CO, United States). Atherton Press, New York, NY, United States, 63–119.

[66] Donald A. Norman. 1988. *The psychology of everyday things.* Basic Books, New York, NY, United States.

[67] Donald A. Norman. 2008. Signifiers, Not Affordances. *Interactions* 15, 6 (November 2008), 18–19. https://doi.org/10.1145/1409040.1409044

[68] G. R. Oldham and A. Cummings. 1996. Employee creativity: Personal and contextual factors at work. *Academy of Management Journal* 39, 3 (1996), 607–634.

[69] Jonas Oppenlaender, Rhemaand Linder, and Johanna Silvennoinen. 2023. Prompting AI Art: An Investigation into the Creative Skill of Prompt Engineering. arXiv:arXiv:2303.13534

[70] François Osiurak, Christophe Jarry, Philippe Allain, Ghislaine Aubin, Frédérique Etcharry-Bouyx, Isabelle Richard, Isabelle Bernard, and Didier Le Gall. 2009. Unusual use of objects after unilateral brain damage. The technical reasoning model. *Cortex* 45, 6 (2009), 769–783.

[71] François Osiurak, Christophe Jarry, and Didier Le Gall. 2010. Grasping the affordances, understanding the reasoning: toward a dialectical theory of human tool use. *Psychological review* 117, 2 (2010), 517.

[72] Rivka Oxman. 2006. Design and theory in the first digital age. *Design Studies* 27, 3 (May 2006), 229–265. https://doi.org/10.1016/j.destud.2005.11.002

[73] Martin Pollák, Marek Kočiško, and Jozef Dobránsky. 2021. Analysis of software solutions for creating models by a generative design approach. *IOP Conference Series: Materials Science and Engineering* 1199, 1 (2021), 012098.

[74] Kaare S. Poulsgaard and Lambros Malafouris. 2020. Understanding the hermeneutics of digital materiality in contemporary architecture: a material engagement perspective. *AI & Society* 38 (8 2020), 2217–2227. https://doi.org/10.1007/s00146-020-01044-5

[75] Simone Pozzi and Sebastiano Bagnara. 2013. Individuation and diversity: the need for idiographic HCI. *Theoretical Issues in Ergonomics Science* 14, 1 (Jan. 2013), 1–21. https://doi.org/10.1080/1464536x.2011.562564

[76] Casey Reas and Ben Fry. 2006. Processing: programming for the media arts. *AI & Society* 20, 4 (2006), 526–538.

[77] Miguel A. Renom, Baptiste Caramiaux, and Michel Beaudouin-Lafon. 2022. Exploring Technical Reasoning in Digital Tool Use. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, United States) *(CHI '22)*. ACM, New York, NY, United States, Article 579, 17 pages. https://doi.org/10.1145/3491102.3501877

[78] Miguel A. Renom, Baptiste Caramiaux, and Michel Beaudouin-Lafon. 2023. Interaction Knowledge: Understanding the 'Mechanics' of Digital Tools. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. ACM, New York, NY, United States, Article 403, 14 pages. https://doi.org/10.1145/3544548.3581246

[79] Michael Rescorla. 2020. The Computational Theory of Mind. In *The Stanford Encyclopedia of Philosophy* (Fall 2020 ed.), Edward N. Zalta (Ed.). Metaphysics Research Lab, Stanford University, Stanford, CA, United States.

[80] Christoph Richter and Heidrun Allert. 2011. The epistemic role of artefacts in creative design and knowledge creation. In *Proc. 13th International Conference on Engineering and Product Design Education* (London, United Kingdom) *(E&PDE '11)*. The Design Society, Glasgow, United Kingdom, 103–108.

[81] Tom Roosendaal. 1994. Blender.

[82] David Rutten. 1990. Grasshopper for Rhino3D.

[83] Donald A Schön. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-based Systems* 5, 1 (1992), 3–14.

[84] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. 2014. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (2014), 2161–2170.

[85] Herbert Simon. 1988. Creativity and motivation: A response to Csikszentmihalyi. *New Ideas in Psychology* 6, 2 (1988), 177–181.

[86] Herbert A. Simon. 1973. The structure of ill structured problems. *Artificial Intelligence* 4 (1973), 181–201.

[87] Herbert A. Simon. 1988. The Science of Design: Creating the Artificial. *Design Issues* 4, 1/2 (1988), 67–82.

[88] Gilbert Simondon. 1958. *Du mode d'existence des objets techniques.* Éditions Aubier-Montaigne, Paris, France.

[89] Gillian Smith. 2014. Understanding Procedural Content Generation: A Design-Centric Analysis of the Role of PCG in Games. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (Toronto, Canada) *(CHI '14)*. ACM, New York, NY, United States, 917–926. https://doi.org/10.1145/2556288.2557341

[90] Jonathan A Smith. 1996. Beyond the divide between cognition and discourse: Using interpretative phenomenological analysis in health psychology. *Psychology and health* 11, 2 (1996), 261–271.

[91] Jonathan A Smith, Paul Flowers, and Michael Larkin. 2009. *Interpretative phenomenological analysis: Theory, method and research.* SAGE Publications, Thousand Oaks, CA, United States.

[92] SideFX Software. 1996. Houdini.

[93] John Stasko. 2014. Value-Driven Evaluation of Visualizations. In *Proceedings of the 5th Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (Paris, France) *(BELIV '14)*. Association for Computing Machinery, New York, NY, United States, 46–53. https://doi.org/10.1145/2669557.2669579

[94] Robert Sternberg. 2006. The nature of creativity. *Creativity Research Journal* 18, 1 (2006), 87–98.

[95] Ivan E Sutherland. 1964. Sketchpad a man-machine graphical communication system. *Simulation* 2, 5 (1964), R–3.

[96] Dassault Systèmes. 2018. CATIA xGenerative Design.

[97] Jerry O. Talton, Daniel Gibson, Lingfeng Yang, Pat Hanrahan, and Vladlen Koltun. 2009. Exploratory Modeling with Collaborative Design Spaces. *ACM Transactions on Graphics (TOG)* 28, 5 (December 2009), 1–10. https://doi.org/10.1145/1618452.1618513

[98] Grant David Taylor. 2013. "Up for Grabs": Agency, Praxis, and the Politics of Early Digital Art. *Lateral, Journal of the Cultural Studies Association* 2 (2013), 15.

[99] Unity Technologies. 2005. Unity.

[100] Carlota Torrents, Natàlia Balagué, Ángel Ric, and Robert Hristovski. 2020. The motor creativity paradox: Constraining to release degrees of freedom. *Psychology of Aesthetics, Creativity, and the Arts* 15, 2 (2020). https://doi.org/10.1037/aca0000291

[101] Amomymous User. 2019. Playing with knobs and rubber bands. https://www.reddit.com/r/synthesizers/comments/ei8lsi/playing_with_knobs_and_rubber_bands/. Accessed: 2023-09-20.

[102] Mario Verdicchio. 2015. The Role of Computers in Visual Art. In *Proceedings of the 3rd International Conference on History and Philosophy of Computing (HaPoc '15)*. DHST/DLMPST Inter-Division Comission for the History and Philosophy of Computing, Springer, Berlin, Germany, 287–299.

[103] Eric von Hippel. 1986. Lead Users: A Source of Novel Product Concepts. *Management Science* 32, 7 (July 1986), 791–805. https://doi.org/10.1287/mnsc.32.7.791

[104] Judy Wajcman. 2010. Feminist theories of technology. *Cambridge Journal of Economics* 34, 1 (1 2010), 143–152. https://doi.org/10.1093/cje/ben057

[105] Richard Woodman, John Sawyer, and Ricky Griffin. 1993. Toward a Theory of Organizational Creativity. *Academy of Management Review* 18, 2 (1993), 293–321. https://doi.org/10.5465/amr.1993.3997517