Miguel Carvalhais
i2ADS & Faculty of Fine Arts,
University of Porto

Designing (with) Computational Objects

From Metamedia to Metaenvironments

Keywords
— design; computation; artificial intelligence; creativity; computational objects.

DOI
— 10.14195/1647-8681_14_7

We live in a regime of computation, a post-digital condition in which we coexist with a technological unconscious that surrounds us and saturates our lives. This paper looks at how this affects us as citizens, and how it transforms our practices as designers, architects, artists, and creators of things. Informed by design, it examines how computation impacts things and spaces—from tools to media, from architecture to environments—and how its affordances breed new objects that are ontologically at odds with the non-computational things and spaces we have grown accustomed to. Through a critique of current tools built with machine learning, this paper enquires how we can negotiate authorial positions in this algorithmic world, and how working within computation reshapes the core tenets of design, or even casts a light onto what those have really been all the time.

The (Computational) World

In his 1974 book *Computer Lib/Dream Machines*, Ted Nelson hinted at the impact that computers, which were transforming into something more than mathematical tools, would have in our lives.[1] By 1974 the usage of computers was starting to broaden beyond what had been its domain in universities, large corporations, or armies, and at the dawn of personal computing, Nelson understood that this could lead the world to revolutionary transformations.

As Nelson said then, "we live in media, as fish live in water," but fifty years later, transformed by computers and computational technologies, our media are now digital, and our world is saturated with computation.[2] We live in a post-digital condition, a regime of computation in which we coexist with a technological unconscious that is formed by objects, by the software embedded in them, and by their networks, and that intersects with us in ways that are often subtle and almost unnoticeable.[3] Computation became a fundamental infrastructure of contemporary life, often mysterious due to its complexity and opacity, in spite of the quasi-permanent veneer of user-friendliness and immediacy.[4] The technological unconscious is perhaps the most salient aspect of the "massively interconnected system of technology vibrating around us," which Kevin Kelly calls the *technium* and that Manuel DeLanda, following Deleuze and Guattari, defines as the *machinic phylum*: a negentropic process of nonorganic life, self-organising processes, and flows of matter, energy, and information.[5]

The technium is an accelerant of hominisation, the process of emergence of the human, acting at the level of social organisation and in the development of physical and cognitive extensions.[6] Starting with the usage of computers as media, it is now leading to the emergence of computational spaces, as computation and computational technologies change their location and time, going from being very localised and sporadic experiences to becoming a ubiquitous and constant state.

All objects are becoming computational, in a spectrum that spans from *coded objects*, where software becomes a part of the desired performance, through *coded infrastructures*, *coded processes*, and *coded assemblages*, a range of *code/spaces* that transform "the conditions through which society, space, and time, and thus spatiality, are produced."[7] When computation becomes "mutually constituted" with our objects, spatiality, cognition, cultures, and societies, our very lives start resembling code/spaces too, i.e., objects that do not dispense with computation, and do not even function without it.[8]

And as computation becomes so enmeshed with objects, and these became difficult to individuate from computation, everything increasingly becomes — much as computation, and because of its influence — cognisant, intelligent, protean, and abstract.[9] And everything also becomes increasingly hackable, unstable, unsafe, adversarial,

1 Theodor H. Nelson (1974), "Computer Lib / Dream Machines," in *The New Media Reader*, ed. Noah Wardrip-Fruin and Nick Montfort (Cambridge: MIT Press, 2003), 301–38.
2 Ibid., 306.
3 Florian Cramer, "What Is 'Post-Digital'?," *A Peer-Reviewed Journal About Post-Digital Research* (2013); N. Katherine Hayles, *My Mother Was a Computer* (Chicago: University of Chicago Press, 2005); Nigel Thrift, "Remembering the Technological Unconscious by Foregrounding Knowledges of Position," *Environment and Planning D: Society and Space* 22, no. 1 (2004): 175–90.
4 Julia Velkova and Anne Kaun, "Algorithmic Resistance," in *Skin and Code*, ed. Daniel Neugebauer (Berlin: Haus der Kulturen der Welt, 2021), 57–74.
5 Kevin Kelly, *What Technology Wants* (New York: Viking, 2010); Manuel DeLanda, *War in the Age of Intelligent Machines* (New York: Zone, 1991); DeLanda, *A Thousand Years of Nonlinear History* (New York: Zone, 1997).
6 Pierre Lévy, *Collective Intelligence* (Cambridge: Perseus, 1997); Kelly, *The Inevitable* (New York: Penguin, 2016), 273.
7 We will define *objects* following object-oriented ontology, i.e., in a very "wide sense: an object is anything that cannot be entirely reduced either to the components of which it is made or to the effects that it has on other things." See Graham Harman, *Object-Oriented Ontology* (London: Pelican, 2018), 43. OOO's objects can be material or immaterial, they can be actual, virtual, even imaginary. This term will then cover tools, media, and all artifacts impacted by computation, including organisms and the contexts where they live. Rob Kitchin and Martin Dodge, *Code/Space* (Cambridge: MIT Press, 2011), 13.
8 Kitchin and Dodge, *Code/Space*, 16.
9 Hayles, *Unthought* (Chicago: University of Chicago Press, 2017).

surveillant, and discorrelated.[10] Imbued with computation and software, everything connects to our dataome and contributes to it.[11]

Computation absorbed media and gave rise to *metamedia* that have "degrees of freedom for representation and expression never before encountered" and the capacity to remediate and transform into existing media, as well as media that are yet to be invented, or are physically impossible.[12] As its impact extends to spaces, via computations embodied in robots, objects, or other automata, it gives rise to new, singular *metaspaces*. Not those of fiction, or of product pitches from technology companies, with headsets, or gloves, or low-res, low-polygon-count models.[13] Rather everything and everywhere, analogue and digital, somatic and molar, physical and immaterial, actual and virtual, *metainterfaces* or *metaenvironments* that potentially touch and include everyone.[14]

Computational Objects

Computational objects take many forms and can hardly be described in simple terms. They can be conceptualised from a set of common computational affordances, as Janet Murray did when describing them as being *procedural*, *participatory*, *spatial*, and *encyclopaedic*.[15] This set the groundwork for an ontology that is, much as the computational objects themselves, hierarchical, with procedurality taking a central place as the "defining ability to execute a series of rules."[16]

A computation is a process that follows rules that are both finite and describable.[17] These rules are usually algorithmic and formal, a principle that has been central to digital computation since it was devised by Alan Turing.[18] The realm of computational objects is not limited to digital computations and computing engines, and includes a host of other phenomena that fall under what can be described as analogue computers, or as chemical or biological systems that are able to process information and *compute*, a group that also includes humans. However, as most computational systems in the technium are digital, we will focus the discussion on those.

Because of their procedural nature, computational objects are paradoxically both deterministic and irreducible, i.e., impossible to anticipate or predict. This means that although we may know that a given computation can only converge into a single output or end state that is deterministically dictated by its rules, we may never be able to know what that state is before the computation actually reaches it.[19] At the same time, in some computations, the same deterministic end result can sometimes be arrived at through multiple causal paths, i.e., different sequences of events.[20]

Computational objects are also heavily predicated on ideas of imitation, simulation, and emulation.[21] The universal computing machine that Turing conceived is an apparatus that can read the description of another machine and start performing its operations. This means that

10 Shane Denson, *Discorrelated Images* (Durham: Duke University Press, 2020).
11 Caleb Scharf, *The Ascent of Information* (New York: Riverhead, 2021).
12 Alan Kay (1984), in Brenda Laurel, *Computers as Theatre* (Reading: Addison-Wesley, 1993).
13 Among the fictions we can include the *metaverse* of Neal Stephenson's *Snow Crash* (1992), or the *matrix* in William Gibson's *Neuromancer* (1984) or the eponymous movies. In technological products we can include Linden Lab's *Second Life* or Meta's *Horizon Worlds*.
14 Christian Ulrik Andersen and Søren Bro Pold, *The Metainterface* (Cambridge: MIT Press, 2018).
15 Janet H. Murray, *Inventing the Medium* (Cambridge: MIT Press, 2012).
16 Ian Bogost, *Unit Operations* (Cambridge: MIT Press, 2006); Murray, *Hamlet on the Holodeck* (Cambridge: MIT Press, 1997).
17 Rudy Rucker, *The Lifebox, the Seashell, and the Soul* (New York: Thunder's Mouth, 2005).
18 Alan Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem" *Proceedings of the London Mathematical Society* 2, no. 43 (1936): 544–46.
19 Stephen Wolfram, *A New Kind of Science* (Champaign: Wolfram Media, 2002).
20 Idem., *A Project to Find the Fundamental Theory of Physics* (Champaign: Wolfram Media, 2020).
21 Miguel Carvalhais, *Art and Computation* (Rotterdam: V2_Publishing, 2022).

a universal computing machine can compute anything that is computable, as anything computable is the output of a machine that can be emulated by a universal machine. But by performing as another machine, a universal computer does not simply simulate or emulate it. What happens in universal computers, and in computational objects in general, is that because of the nature of the processes that are developed, we find something that is more than an emulation, more of a *recreation*. When a universal computer is programmed to *act as* another machine, it can be thought of as actually *becoming* the other machine, transforming itself into a different machine that is then able to produce new computational outputs that are identical to those produced by the original machine.

As a computation then emerges from the affordances of its physical substrate — the hardware — and its links with the rules and data that drive it — the software — it becomes something that is neither one nor the other. It becomes a new informational and causal object that is at once strictly reliant on its material and informational substrates, but also independent from them.[22] As an irreducible informational being that is structurally coupled with its context, any new computation develops a new informational pattern, a copy without an original, a *simulacrum*.[23] It is not a copy, simulation, or emulation *of* something, it *is* something.

A computation has agency, can make choices, act, and express intentions, or drives. All of these are, of course, programmed into it in one way or another, but often, we can see manifestations of a computational object's subjectivity when it turns into a black box that escapes the control of its designers and programmers.[24] Furthermore, the context with which the computational object structurally couples, includes not only any matter, energy or information that can be perceived by or acted upon, but also any number of other computational objects or organisms — including humans — that it may be able to interact with. Everyone and anything that interacts with a computation becomes part-processor and part-programmer, and all actions in this mesh of living and non-living, of natural and artificial things, become ontologically equivalent, contributing to and becoming part of the computational object.[25]

Metamedia

Computational media deeply transformed the media ecology and our aesthetic relationships.[26] Although computation had haunted media for a long time, it was only when digital computers became a universal medium, a solvent for all forms of sign production and communication, that we started breaking away from some of the paradigms that had for long shaped our culture.[27] Given a technological context with so many radically new affordances, media evolved into forms that go beyond imitation and remediation.

Computational media are outstanding remediators, that promise a transparent replacement of all media with cheaper, faster, more reliable

22 Max Tegmark, *Life 3.0* (New York: Alfred A. Knopf, 2017).

23 Hayles, "Embodied Virtuality: Or How to Put Bodies Back into the Picture," *Immersed in Technology*, eds. Mary Anne Moser and Douglas MacLeod (Cambridge: MIT Press, 1996), 1–28; Jean Baudrillard, *Simulacra and Simulation* (Ann Arbor: University of Michigan Press, 1994), 1981.

24 Alexander R. Galloway, *Uncomputable* (London: Verso, 2021).

25 Bogost, *Alien Phenomenology, or What It's Like to Be a Thing* (Minneapolis: University of Minnesota Press, 2012).

26 Matthew Fuller, *Media Ecologies* (Cambridge: MIT Press, 2005).

27 For an overview of some of these, see Carvalhais, *Artificial Aesthetics* (Porto: U. Porto Edições, 2016); Friedrich A. Kittler, "The Finiteness of Algorithms," 2007, accessed October 2022, 2018, https://archive.transmediale.de/content/the-finiteness-of-algorithms; Ed Finn, *What Algorithms Want* (Cambridge: MIT Press, 2017).

and accessible alternatives that expand what can be done with media, but also augment and transform our selves, fulfilling the utopian expectations of visionaries such as Vannevar Bush, Douglas Engelbart, or Nelson himself.[28] And computational media mostly deliver on these promises, causing a convergence of media and culture and the ongoing development of *new media* forms out of a richly generative substrate.[29]

More than any previous media forms, computational media are systemic. As Yuk Hui points out, this is not only because they are coherent but also because all media that follow them — and that are based on them — are also essentially computational.[30] The computational becomes the substrate of all media, even of those with which it does not directly intersect. By turning all information into a universal format of representation, we get Manovich's *monomedium* and Kay's *metamedium*; we get something that is amodal, endlessly transcodable, plastic, interactive and mutable.[31]

But the metamedium entails a paradox at its heart. Because it is built upon computation, it cannot avoid expressing computation in some form. Computation is transiency, action, and change; it is transformation of information rather than its preservation and stabilisation. Its very nature is in many ways antithetical to the role it is required to take on as a medium. Computations, and consequently computational media, are natural producers of information and signs, much more than their keepers.

We always expected our media to be neutral — even though we know they can never be — therefore we developed several techniques and strategies to assure that information is stable and endlessly repeatable in the metamedium. But because all information in computational media is stored in the same format and needs to be transcoded into the modalities of output that are used, all signs — letters, images, sounds, etc. — need to constantly be recreated by the medium, created anew every time they are mediated. This is a process that creates ample opportunities for the expression of computation — through phenomena such as discorrelation, glitch, or poor images — and that turns the computational medium into a space of uncertainty and performance where new signs are often created but not mediated, existing only in the liminal space of the computation itself.[32]

Amidst this, we learn to be aware of the agency of computational media, of how alien it is to humans and how it produces phenomena that although genuinely unprecedented, quickly became part of our culture and aesthetics.[33] We learn to never absolutely trust our media, to always be vigilant about their potential to drift, to mutate, to create.

Metaenvironments

Many of these considerations also extend to those objects through which computation, software, and information materialise in the world and act upon it. Whether these objects are immaterial and exclusively

28 Vannevar Bush (1945), "As We May Think," in *The New Media Reader*, eds. Noah Wardrip-Fruin and Nick Montfort (Cambridge: MIT Press, 2003), 35–47; Douglas C. Engelbart, *Augmenting Human Intellect* (Menlo Park: Stanford Research Institute, 1962); Nelson (1965), "A File Structure for the Complex, the Changing and the Indeterminate," in *The New Media Reader*, 133–146.

29 Henry Jenkins, *Convergence Culture* (New York: New York University Press, 2006, 2008); Lev Manovich, *The Language of New Media* (Cambridge: MIT Press, 2001).

30 Yuk Hui, "On the Cosmotechnical Nature of Writing," in *The New Alphabet*, ed. Bernd Scherer, 56-71 (Berlin: Haus der Kulturen der Welt, 2021).

31 Something that is not tied to specific modalities of representation, or to any type of representation at all, with perhaps the exception of the voltage fluctuations that represent bits at the level of hardware, but even those are abstract and can thus take many different forms.

32 Jordan Schonig, "Cinema's Motion Forms," (PhD Dissertation, University of Chicago, 2017); Hito Steyerl, "In Defense of the Poor Image," *E-flux*, no. 10 (2009); Rosemary Lee, "Aesthetics of Uncertainty," *xCoAx 2019*, Milan, 2019; Lee and Carvalhais, "Net Art and the Performance of Images," in *The Web that Was: Archives, Traces, Reflections* (Amsterdam: University of Amsterdam, 2019); See also Trevor Paglen's concept of the invisible image in "Operational Images," *E-flux*, no. 59 (2014), developed after Harun Farocki's discussion of operational images in "Phantom Images," *PUBLIC* 29 (2004): 12–22.

33 Denson, *Discorrelated Images*, 107.

informational, or material and physical, characteristic traits of computation get instilled into them and are often manifested.

Computation is informational, dependent on structural couplings developed through exchanges of information. With or without what we identify as *interactions*, computational objects continuously react and adjust to their environments. In this sense, computational objects are not so much what we encounter but rather what we make with them. They are dependent on feedback loops and the information exchanged is not only functional but also paramount for computational aesthetics, for the ways how we perceive and relate to computational objects. Information is what allows humans to interpret computational objects and to develop models of their causal processes, *theories of the system* that will assist them in acting together.

This makes computation performative, as it only happens when it is enacted, when its infrastructures set it in motion, and it emerges from hardware, software, and context. It is in this enactment that computation becomes, as a performative object that is not only encountered in time but is composed of time, composed of a temporal execution of code at extremely small scales, that generates what can be seen "as a new form of time," where things are "the very medium through which time is modulated."[34] Through this, at every step, a computation loses some futurality but this is offset by the simultaneous restarting of the computational cycle and consequent accrual of potential.[35] Thus, the coupling with a computational object is an experience of immersion in a spatiotemporal experience that requires modes of engagement that are subjective, situated, and enactive, where we explore fields of possibilities for the computational object and all the things it expands to.

Computational objects are experiential. Through their irreducibility and discorrelation, they become particulars, unique entities with which we engage as individuals, and that make us aware of the acts that are involved in their perception.[36] This engagement with computational objects is predicated on the development of theories of the system that confront our world of experience, our *Umwelt*, and make us reconceptualise it by perceiving the world through a cybernetically expanded sensorium and sharing a *technological Umwelt*.[37]

Computational objects are situated; they are concrete assemblages that exist in a specific time and space and that veer towards divergence, confronting us with events that happen to us and now. They mediate and transform our experience of space and become hybridised with our environments, combining them in ever-new ways and creating singular relationships with their inhabitants, be they organisms or other systems. They are agents, acting on and generating information. They are planes of immanence that bleed into the physical world and touch us, but that are also transformed by the relationships we develop with them. They thus function like languages and interfaces, placing us in an ergodic space of cooperation and *com-putation*.[38]

34 Ibid., 228.

35 *Futurality*, or *future future* is, for Morton, the possibility that things can come to be different, their potential for an irreducible and unpredictable to-come that characterises and realises beings. See Timothy Morton, *Being Ecological* (London: Pelican, 2018).

36 Alva Noë, *Strange Tools* (New York: Hill and Wang, 2015).

37 Lee, "The Limits of Algorithmic Perception" *Politics of the Machines: Art and After*, Copenhagen 2018. https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/EVAC2018.0

38 Andersen and Pold, *The Metainterface*; Espen J. Aarseth, *Cybertext* (Baltimore: Johns Hopkins University Press, 1997). "The term computation itself comes from the Latin *computare, com-* 'together' and *putare* 'to reckon, to think or to section to compare the pieces.'" David M. Berry, *The Philosophy of Software* (Basingstoke: Palgrave Macmillan, 2011), 10.

Computational objects do not encode forms and signs but rather data and rules, and their perceived stability and permanence is nothing more than a manufactured illusion. Computation is never stable, it is defined by permanent change, instability, and irreducibility. Any sign or behaviour we can perceive in a computation is continuously constructed and mediated. In computational objects, mediating is enacting, and ontology and phenomenology become entangled in ergodic acts that reveal agency. Time is not only where computation takes place but is also something that emanates from it. A computation is a succession of discrete steps that through processes of actualisation and differentiation bring a formal and algorithmic past of code into a futural *not-yet*. Because of this, morphology can never be stabilised, as that would require stopping computation and hardening a thing in its past and its appearance, not its being. Computational objects are machines of uncertainty and hermeneutics, they are gnarly systems that stand in between indeterminacy and seemingly stable configurations that are sometimes formed in fleeting pockets of reducibility.

Therefore, computational objects are highly improvisational, with their real-time events existing in a rule-bound space, and their significance depending on the exploration of its possibilities. Our interactions with computational objects are also improvisational acts, oriented to goals and tasks that are met through the engagement with a system of rules that acts as its framework but is unpredictable and forces us to improvise. They become experimental because of this uniqueness and unrepeatability. They are processes that generate action, less concerned with prescribing forms or signs but rather with creating phenomena from which forms or signs may result. They thus raise questions that can only be formulated through processes and that depend on enactment.

Computational objects are immanent, conjuring their own space and time when they are enacted and materialise. They exist across planes, and when computation stops, it does not disappear but withdraws to a space of rules, code, storage, memory, and immanence, returning whenever it is enacted again.

Through enaction, computational objects become theatrical. Computation is not formalist or literal and whenever we engage with it, we experience computational objects through mediations that demand our involvement. Computational objects are dependent of, and invested on, the circumstances in which they can engage with others, and of being encountered in a context that includes the other. This creates situations that are outside the boundaries of any of the systems and that derive their meaning from the relationships between all the engaging parts, from a theatrical enactment between the sensorial and material and the informational and computational.[39]

Computations are defining for computational objects, and if they stop, the objects change states and become something quite different.

39 Harman, *Art and Objects* (Cambridge: Polity Press, 2020).

But they do not happen in their hardware or software. Computations are not the hardware but rather how the hardware acts and processes information. They are liminal, happening in between objects and in between planes in objects. They are an abstract informational phenomenon that, although tied to physical mechanisms, is not physical itself, and exists only while it is enacted, creating a temporal space that is gone once the computation ends. Computational objects are spectral, uncanny things that fluctuate between real and unreal, hardware and software, actually and memory.

In their complexity and fleetingness, computational objects are never fully present or wholly perceivable. No single moment of a computation is the computation itself, and often there is no completion or end-state we can anticipate arriving at. Computations are objects to which we may attune, that we can discover, and with which we can collaborate. With perhaps the exception of the most simple, predictable, and reducible computations, most computational objects become "massively distributed in time and space relative to humans" and are difficult to perceive directly.[40] Their complexity and diversity can be well described by the criteria that Timothy Morton developed for *hyperobjects* such as the biosphere, planets, global warming, or other systems that are fundamentally characterised by viscosity, nonlocality, temporal undulation, phasing, and interobjectivity.[41] They are sticky and adhere to things, becoming entangled with them in irreducible and uncontrollable ways. They are perceived through local manifestations that are not the computations themselves; they are a mesh of which we, and their local manifestations, are parts. They do not need to permanently fit our four-dimensional space-time but may exist across other dimensions, spaces, and scales, and therefore phase in and out of this one. As they compute, they radiate temporality and phase in and out of forms which we can perceive. All of this happens through the development of relationships within this mesh, between objects, processes, their links, and the gaps within them. The hyperobjectual nature of computational objects dissolves causes, signs, and information that are to be found among and in-between objects, an in-between that "is not 'in' spacetime" but rather "is spacetime."[42]

Irreducibility also contributes to computational objects having a creative potential. Computational objects can explore conceptual spaces, and transform them, doing things that at the very least *seem* creative.[43] The main activity of any computation is to continuously create itself, a process during which, creativity is manifested in the changes of how objects act on their environments and on themselves.[44] The potential for creativity arises from a spectral and ecological presence that is not past (in the code), present (in the moment-to-moment actions), or future (irreducible) but that seemingly exists outside of time, hovering above the computational substrate and permanently building new relationships.

This contributes to the subjectivity of computation and computational objects. On the one hand, by witnessing action and

40 Timothy Morton, *Hyperobjects* (Minneapolis: University of Minnesota Press, 2013), 1.
41 See Morton's *Hyperobjects* and Carvalhais's *Art and Computation*.
42 Morton, *Hyperobjects*, 81.
43 Margaret A. Boden, *The Creative Mind,* second ed. (London: Routledge, 1990).
44 Joanna Zylinska, *AI Art* (Ann Arbor: Open Humanities Press, 2020).

identifying agency, we project subjectivity onto artificial systems that can behave in complex ways and cognise. On the other, any choice of behaviour of a computational object, is not, as we have seen, strictly in its past, as a mere rule encoded in software or hardware, but is rather something that happens moment to moment, that is dependent on variable contexts and that can be seen as the machine's own choice, irreducible and subjective.[45]

Aesthetics

From the affordances of computational objects arise new types of relationships with humans and other things. The aesthetic relationships with computational objects are quite different from those developed with other things, media, or tools, perhaps with the exception of those we develop with complex organisms. In these we develop procedural readings, through a computational gaze and computational interpretations.[46] These ergodic experiences allow us to discover computations, to intuit their *allegorithms* and get to know the systems intimately, to the point where the computations in computational objects, other computations that simulate them—such as the models we develop—and those computations in the broader context where both the computational objects and ourselves exist, may be indistinguishable.[47] The aesthetic relationships with computational objects are relationships of interaction, behaviour, and modelling. They are relationships of anticipation and appropriation of the computations and their multiplication through theories of the system that enact new computations that will in their turn develop relationships with us and, through us, with the environment and the computational objects. Computational aesthetics is an aesthetics of futurality, of trying to anticipate the behaviours and choices of a computation through modelling its phenomena and enacting new future futures.

Designing for the Meta-

How does such a shift in our tools, media, and environments affect how we design? Christopher Alexander stated that the "ultimate object of design is form," but what designers design nowadays are behaviours from which form and agency emerge.[48] Designers create computation; all else follows from there.

Objects and spaces are no longer static, they are actant, and thus transform the nature of design. In the early days of the metamedium, Murray discussed procedural authorship by describing the author's role as "writing the rules by which the texts appear as well as writing the texts themselves." These rules would govern the involvement of interactors and the responses from the system, "the properties of the objects and potential objects in the virtual world and the formulas for how they will relate to one another."[49] Authors would therefore create "a world of narrative possibilities," fulfilling a role comparable to that of "a choreographer who supplies the rhythms, the context, and the set of steps that will be performed," and allowing the

45 Friedrich A. Kittler, "The Artificial Intelligence of World War: Alan Turing," in *The Truth of the Technological World*, ed. Hans Ulrich Gombrecht (Stanford: Stanford University Press, 2013), 178-94.

46 Carvalhais, "Breaking the Black Box," *Artificial Intelligence and the Arts*, eds. Penousal Machado, Juan Romero and Gary Greenfield (Berlin: Springer, 2021), 347-62.

47 Galloway discusses a level of interpretation of a computational system where one allegorically internalises them with the realities of the broader context: "video games are, at their structural core, in direct synchronization with the political realities of the informatic age," which leads him to suggest that these objects are not only algorithmic but also *allegorithmic*. Galloway, *Gaming* (Minneapolis: University of Minnesota Press, 2006), 91; McKenzie Wark, *Gamer Theory* (Cambridge: Harvard University Press, 2007), 30.

48 Christopher Alexander, *Notes on the Synthesis of Form* (Cambridge: Harvard University Press, 1971), 15.

49 Murray, *Hamlet on the Holodeck*, 152.

interactor to use that repertoire "to improvise a particular dance among the many, many possible dances the author has enabled."[50]

Following an ontological understanding of computational objects, we can also consider the computational objects themselves among the dancers, and Murray's improvised dance as being mutually constituted by human and computational objects in tandem. It is more than a collaboration, it is coprocessing.[51]

Computational objects exist in a duality between their *subfaces*, that are procedural and largely hidden from view, and the sensorial *surfaces* through which they manifest to the world.[52] The surface is not itself computational, but all its signs and behaviours are driven by the subface, a black box to which we have no access but that reacts to what happens in and with the surface. Subface and surface are one, manifesting as different strata of an object. A consequence of this dual nature is that, as designers, we do not deal with static materials any more but rather with objects that cognise, that have some sort of a mind, that have drives, desires, and a psychology.[53] Objects that we can model and that also model us.[54]

These objects are autonomously intelligent, and their capacity to solve problems across different domains or to develop their own models of the world and of the things in it does not exist merely in the past, i.e., as a transfer of intelligence from designers or programmers into the systems.[55] This is a view that *good old-fashioned* approaches to artificial intelligence often espoused, when trying to formalise decision processes for complex and creative behaviours, but that both more recent approaches to modelling intelligence and cognition do not, as they are more prone to understand intelligence as an emergent process that is at once embodied and substrate independent.

So, perhaps, more than analogies with choreography, we should think about design as dramaturgy and as the creation of living structure.[56] What do we design then? We create behaviour, action, and agency, and these exist within what we can describe as a *phase-space*, a *latent space*, or even a *conceptual space*, to use a term with connections to Margaret Boden's work on creativity.[57] Whatever name we prefer, the significant characteristic of these spaces is that they are shaped by the affordances and constraints of a system and contain all of its possible states.[58] Their exploration allows the analysis of affordances and constraints, the discovery of new states within the spaces, and even the transformation of the spaces through a metamorphosis of the system.[59] Knowledge of these spaces allows the development of a perspicuous overview that translates into understanding them and the world they are a part of.[60]

This is a rather indirect way to design. Instead of making plans through self-conscious methodologies that abstract problems and solutions, the designer operates by programming. This may mean coding computational objects, interacting with them, or shaping computation by any other means. Regardless of the approach, the designer does not

50 Ibid., 153.

51 Aarseth, *Cybertext*.

52 Frieder Nake and Susanne Grabowski, "The Interface as Sign and as Aesthetic Event," in *Aesthetic Computing*, ed. Paul A. Fishwick (Cambridge: MIT Press, 2006), 53–70.

53 *Mind* can be defined, in a fundamental sense, as something whose central activity is to think, changing inputs into outputs trough informational processes. A mind is an "action noun," something that responds, transforms, acts, adapts. A mind is not a thing, but a dynamic system defined by change. For more on these definitions, on mind's embodiment across multiple systems and organisms, see Ogi Ogas and Sai Gaddam, *Journey of the Mind* (New York: W. W. Norton & Company, 2022); Psychology is defined as a set of internal states that the system may or may not be aware of but that contribute to shaping its behaviours and drives, and even, ultimately, the system itself.

54 When we develop a theory of the system of any computational object we have contact with, we model not only its surface but also its behaviours and, to a certain extent, its internal states, also developing theories about its mind and its outlook of the world, its Umwelt. See Carvalhais, *Art and Computation*.

55 As Ada Lovelace and others have suggested. See Boden, *Creative Mind*, 16.

56 Siegfried Zielinski, *Deep Time of the Media* (Cambridge: MIT Press, 2006).

57 The term *latent space* is now common in contexts related with machine learning, such as generative adversarial networks or large language models, and has been used in this context for a few decades. Herbert Franke, the early computer artist, discussed the variability of interactive art and images as the activation of different image variants and sequences that are latent images discovered by users through "something like a journey of exploration through a world an artist has designed." See "The Expanding Medium: The Future of Computer Art," *Leonardo* 20, no. 4 (1987): 335–38.

58 We can think of constraints as negative affordances, and vice versa.

59 Developing something akin to what Boden would call *transformational creativity* and building a new conceptual space that is potentially packed with new, original, and valuable states. This amounts to a transformation of the computational object into a different object by transforming the computational machine within into a new machine.

60 Alva Noë, "Experience and Experiment in Art," *Journal of Consciousness Studies* 7, no. 8–9 (2000): 123–35, 131.

shape form or space directly but rather builds rules that then breed latent spaces. Forms, if any are produced, will be found there.

The latent space will also define horizons of action and of intent for humans and any other systems that cooperate with the computational object.[61] The designer thus shapes allegorithms and traversal functions, mechanisms to explore and discover these spaces; to verify, validate, and trust them, but also to get lost in them.

Working *with* and *for* computational objects, the designer helps to develop systems that seem alive and that can sometimes even directly involve other living systems as their constituents. Thus, *life* may become something more than a metaphor to describe aspects of cognisant computational objects.

Because of this, we discover that the tools with which we work are also the object of our work, and that, furthermore, also act on us. Our tools influence our cognition and our thinking habits.[62] To paraphrase McLuhan, we shape them, and they shape us back. Computation becomes a unified substrate for everything that includes the tools with which we develop computation and the computations we develop with them. Both are equally malleable and shapeshifting; both have the potential to replace and automate our work, something that we gladly allow them to do, sometimes uncritically.

We should not forget that the potential for openness in computational objects can be narrowed by blunt tools, and that, to find value amid the vastness of these latent spaces, we need those tools to not constrain exploration or limit our expressive potential.

An interesting recent example of this phenomenon can be found in image generation systems such as DALL·E 2 or Stable Diffusion, and in how they ultimately constrain our very notion of *image*, through the imposition of a taxonomy, of conceptual models, and tools that narrow what can be made, or what can even be thought about, with them.[63] These systems are machine learning models that can generate a large array of images in a variety of styles from the interpretation of descriptions, or *prompts*, expressed in natural language. This linguistic interface is the main tool they give us access to for the exploration of their latent spaces, allowing us to prompt the system for images of, e.g., "an astronaut, riding a horse, in a photorealistic style," to directly quote one of the examples in the DALL·E 2 website.[64]

There is certainly an interesting spectacle to a computer's interpretation of a prompt and the ensuing generation of an image or set of images.[65] However, the tools that are used to conjure such images sometimes narrow a system's potential. They can make it difficult, even impossible, to express and mediate drives and needs, or to capture anything about them at all.[66] One of the reasons for this is the fact that natural language is often not a good resource to express a design problem or to describe what we are searching for in a latent space. A consequence

61 Brian Upton, *The Aesthetic of Play* (Cambridge: MIT Press, 2015).

62 Edsger W. Dijkstra, "The Humble Programmer," *Communications of the ACM* 15, no. 10 (1972): 859–66.

63 This happens even before any arbitrary limitations built upon the systems, as e.g., not generating faces for celebrities, not producing certain types of images, not responding to specific keywords, etc.

64 https://openai.com/dall-e-2/

65 Robin Sloan points to how most presentations of outputs of these systems include not only the textual or visual output but also the prompt that led to it, hypothesising that the main aesthetic pleasure in these systems does not reside in the outputs themselves but rather on the spectacle of the interpretation that is developed by the computer. See "Notes on a Genre" 2022, accessed September 2022, https://www.robinsloan.com/lab/notes-on-a-genre/

66 Sometimes they make it impossible to even articulate seemingly mundane ideas because of the origin of the datasets used, their contents, and how they were collected. These result in patterns in generated images that are caused by limitations in training data, biases in the outcomes or what Eryk Salvaggio calls "*reductive* system interventions," or *censorship*. See "How to Read an AI Image: The Datafication of a Kiss" 2022, accessed October 2022, https://cyberneticforests.substack.com/p/how-to-read-an-ai-image.

of this is what has become known as *prompt engineering*: methods that try to evoke processes by almost transmuting natural language into abstract constructions that, despite still being composed from recognisable words, start losing the resemblance to natural language.

Programming languages are of course artificial, even if they often resort to words or constructions that are shared with natural languages. They do so, however, by imposing strict specifications that lead them to be less prone to the ambiguity, vagueness, and indetermination of natural languages, to achieve preciseness through tight lexical, semantic, and grammatical control. Because of this, programming languages can be better tools than natural languages to form and explore latent spaces, and of course, no single language can be seen as the ultimate tool for all processes. A reason for this is that in these contexts, natural languages are used as interfaces between coded computational domains that may be easier to explore with artificial languages. Another is that a poetic and creative level of language that explores ambiguity and metaphor is still beyond the reach of current computational systems, and creative ambiguity is a dialectic process that requires all players to engage in it. Perhaps what designers need then, when designing computational objects, are not generic general-purpose tools, using natural or artificial languages, but rather bespoke tools that are themselves developed as part of the design process itself.