Design
Arts
Médias

Exhibit A : Exposing software

Kim Sacks

2020-12-01

Designer/Docteur, ATER à l'Université de Paris 1 Panthéon-Sorbonne
Membre de l'Institut ACTE, Axe « Design, Arts, Médias »

Abstract

This paper addresses the following questions : is *software-based design* hiding something ? And if that is the case, what is it hiding ? And from whom ? Machines are paradoxically legible and unintelligible : they display their visual effects whereas the cause for these effects is hidden in a network of entangled inter-dependencies, generating a system that presents only a fraction of the broader puzzle. By analyzing trustworthiness, didactics and ambiguity of system-based practices, this essay takes a look at art and design pieces by exploring their internal mechanisms in order to see what kind of strategies these pieces are hiding and/or exposing to their audience.

Résumé

Ce texte aborde les questions suivantes : le design basé sur des logiciels cache-t-il quelque chose ? Et si c'est le cas, que cache-t-il ? Et de qui ? En étant paradoxalement lisibles et inintelligibles, les machines affichent leurs effets visuels alors que la cause de ces effets est cachée dans un réseau intriqué d'interdépendances, générant un système qui ne présente qu'une fraction d'un puzzle plus large. En analysant le caractère de confiance (*trustworthiness*), l'approche didactique et l'ambiguité pratiques s'appuyant sur des systèmes informationnels, cet article tente d'examiner des œuvres d'art et de design en explorant leurs mécanismes internes afin de voir quelles sont les stratégies qu'elles cachent et/ou qu'elles exposent au public.

Introduction

The title of this essay is intended to lead the reader towards one specific set of questions : is *software-based design* hiding something ? And if that is the case, what is it hiding ? And from whom ?

Most obviously, an exhibit is meant to expose something, whether it be a piece of artwork or interactive design. But expose in what sense of the word ? Are we trying to strip down an object in order to reveal something that we do not see at first ? The word « exposition » itself has shifted its common understanding : the action of putting something into view in order to give an explanation from a knowledge perspective. From a more contemporary consideration, it seems that art exhibitions have split *explanation* and *display* into two distinct aspects of expositions. In this context, artistic pieces could be considered as exhibits : is artwork the *exhibit A*, being an artwork that *shows* but doesn't *demonstrate* ? Or maybe, does it demonstrate its failure to *expose* its own methods of conception, the process of its own creation, the program enabling the artwork to operate ?

Looking at a painting, one could infer what kind of medium was used : oil painting, inks, brush strokes, even possibly the approximate date the artwork was produced, speculate on the use of live models etc. But when facing a *program*, these deductions are far less obvious – and even sometimes quite impossible. If one is shown a data-driven interactive data-visualization, can we understand the inherent protocol ?

> *« But to conceptualize contemporary aesthetics, we have to confront the ways new media push artistic practice into a systems-based, codependent relation with their conditions of* use *and* discourse*, not merely their formal properties or their capacity to function as social signs in a semiotic mode. Aesthetics is transformed, hybridized, by the challenges of mediation as a central feature of artistic work.*[1] »

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

The specificity of these *system-based practices* is the entanglement of *use* and *discourse*, of *knowing* and *doing*, of *exposing* and *displaying*. Machines are ambiguous. We cannot separate software from hardware, the internal logic relies on the use of machines. And the generated knowledge relies on their technical implementations. These machines execute the program, placing them both as the process and result, cause and effect so to speak. Therefore, one should not apprehend the resulting operation of software without understanding the underlying mechanisms that make it happen, that cause it to appear on screen. As Alexander Galloway reminds us, not unlike a framework introducing a standard governing diplomatic agreements, protocol is not only a technical paradigm guiding the interaction between parties[2]. It is what is defining the relation of parties, agreed upon prior to its implementation. Understanding that protocol is what enables systems to communicate on a standardized basis, it seems as if there is no such protocol that exists in the reception/exposition of software in the context of art/design exhibits. At least, this type of protocol is not explicit ; if a tacit agreement exists, it is one that is dictated by the context of art reception, vastly compounded by the aesthetic reception of visual information and knowledge. No rules are defined that would provide a pattern of legibility for software and the underlying information it presents to its audience. Does that mean that there is no such governance in this context ? Most likely not. But the inter-operation still exists, and calls for guidelines, especially in the context of exhibits. All the more so because legibility is one of the main claims of data-driven design. How does one make data-driven design legible, given that this driving force is what gives meaning to its design ? How does that aspect of design come through in the final object presented to a given audience ?

The leeway that art and design provide, to both producers and curators, leaves a door wide open to the utilization of different strategies. What we will try to explain is the fact that these strategies are not, necessarily, ill-founded or manipulative, but rather an absolute necessity because of the technical specifications of technology oriented design. Code is to design what medium is to painting : it is both the process and the end result. Except that this very particular status provides numerous technical possibilities conferring a potential for clear legibility and/or opaque access to the inherent protocol.

Brief disclaimer : I am not trying, through this paper, to call for exposing bad behaviors, or bad endeavors. I couldn’t claim to be the judge of that, nor even would I claim that one should judge *software-based design* exclusively on the technical protocol. I am not trying to critically analyze one specific art/design piece based on its interpreted significance. This would be a different and fascinating subject. Neither am I trying to say that these pieces are faking something, quite the contrary. Nothing is fake about the visual perception of these pieces. Rather, I am exploring the limit of the audience’s possibilities regarding the understanding of the technical implementations. Creating art with technology is most evidently not a neutral undertaking. What I am trying to do is to look at the motives of technical and artistic ambiguity.

Nonetheless, on one hand, I would like to argue that the strategies used and that we will look into, help to define the perception a spectator will have. They shape the way we accept and interact with the given piece, they create a framework and determine the trust we ought to have in its *authenticity*, both from an aesthetic and informational perspective. On the other hand, I would also like to point out that here lies the paradox of exhibiting software. It relies simultaneously on the primary feeling of distrust (because of the fact that software logic *eludes* our first contact understanding) and on the transparency it is meant to shape (because of the *attractiveness* of digital complexity as a whole). Therefore, we cannot circumvent the hidden aspects of *software-based design* without taking into account the fact that, at its core, this question is about accessing knowledge through the effect of executed programs.

Software has some particularities that non system-based art forms do not have : it relies on computer programs and is manufactured using technology that has come to be so complex that even the most knowledgeable computer scientist wouldn’t necessarily grasp all its internal logic. Most importantly, once exhibited, the question that remains unsolved is how does one break the surface of software in order to reach the meaning, exposing the fragmented interiority of the program ?

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

Here are the key discussion points and hypotheses we will develop in this paper :

1) We speculate that, when an artwork is based on software and therefore relies on the execution of a program, the meaning of the piece cannot be dissociated from its technical implementation, its materiality. Most notably, live data-driven interactive design aims to transpose its protocol into a shape that enables clear access and legibility to the given data. By reversing this logic, if a piece of artwork dissimulates its technical implementation, the access for the spectator to the meaning of the said artwork is difficult, to say the least. By doing so, I think that software-based design hides – voluntarily or not – its own meaning behind complexity and the resulting effects are decoupled from its authenticity. The consequence of this decoupling is an increase of distrust and defiance.

2) The context of exhibitions participates in this defiance : the reception of a piece of software will largely depend on the means of exposition. The premises when entering an exhibition isn't an expectation of authenticity concerning the proper implementation of a certain protocol. Rather, it is one of aesthetics, it is not one of knowledge but one of experience. Because of this, the expectations can be very different if we were to compare them to, let's say internet based data-visualization. While this is true, I think exhibitions also provide a perfect context for a didactic approach to knowledge, such knowledge being both the meaning of the piece, and the protocol itself. We believe that this didactic potential is what could enable the re-coupling of cause and effect, of meaning and protocol.

3) Software needs to account for all possible events and technical randomness that could occur while the software is running. In the exhibition context, the software must run continuously, not unlike a website. From this standpoint, designers and artists who wish to implement *functional* pieces need to project all the probable outcomes : bugs, internet access availability, etc. In order to solve these potential technical glitches, strategies can be put in place, fail-safes such as offline backups of databases. Multiplying these strategies is sometimes chosen as a way to guarantee that the exhibit audience will be able to experience the piece in its functional behavior, and not to be confronted with a blank screen. While most data-driven design pieces resolve the technical challenges with a technical implementation, we have seen artists who specify their own strategy via a simple explanation, as in the example of the 2009 work by Samuel Bianchini, *AllOver*. Even if this explicit approach to exposing the artists' strategies help to grasp the technical implications of data-driven design, it doesn't totally alleviate the ambiguity existing between a perceptible functional program and the embedded strategies. In doing so, we hypothesize that the technical specificities of programs enable this ambiguity, but also offer favorable conditions for the exposition of technical choices, be they discursive or strategic.

Most obviously, we will analyse a certain number of examples in order to show that exhibiting design can simultaneously imply exposing internal technical specificity and providing legible knowledge from an aesthetic experience.

1. Material trustworthiness

Our first hypothesis is the inherent duality of data-driven design : it sits in between machines and humans. It claims to bridge the gap from unintelligible machines via visual interfaces to the users/audience. Andy Kirk describes data-driven visualization with the following definition : « The visual representation and presentation of data to facilitate understanding.[3] »

His clear statement at the opening of his famous book illustrates the structure of the relationship from data to knowledge, and as a consequence, from machines to users. Visualization is (or aims to be) a process of exposing hidden patterns of data and transforming it into an understandable representation (or presentation). In other words, visual representation is meant to expose the underlying data and make it legible. From Kirk's point of view, the design operation is what either makes this bridge hold or collapse. The role of the interface creator would be to manufacture a program that could suggest to its users its trustworthiness, what the author calls *trustworthy* design : « The reliability, consistency and functional performance of a visualisation is something that

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

influences the perceived “trustworthiness”. Does it do what it promises, and can the user trust the functions that it performs ?[4] »

Using this as a starting point, I would comment that the promise of reliability could be thought about from multiple different perspectives :

Firstly, the context radically changes the way we perceive the data-driven object, and therefore, it dictates our evaluation of its trustworthiness. The material trustworthiness is not something uniform that all pieces handle uniformly. If the creator’s aim is to give access to a purely functional program, and perhaps this is Kirk’s perspective, then the trustworthiness of the device, its interface and visualization, is the absolute essential binding relation between an audience and the perceived piece. On the other hand, in the exhibition context, matters are not as binary, strictly opposing aesthetics and functionality. One could argue that the point of exhibiting data-driven design in an exhibition, is to forgo the necessity of function and mainly provide an aesthetic experience. While this is true, to some extent, it simultaneously enables the use of machines in all possible scenarios, inhibiting the potential for a common protocol of perception.

Secondly, the machine itself, independent of the interface, is something the audience perceives with preconceived trust, distrust, or even more likely an ambivalent mixture of both, depending on the specific individual’s culture and background. One doesn’t naturally trust machines, it is something that evolves, changes, because it is conditioned by many factors quite impossible to enumerate (and such is not our goal). This makes the exposition of data-driven design an upstream journey towards trustworthiness. Or at least, if the goal is the one outlined by Kirk, then it should be thought of in this way. But the goal seems not always to be *trust* but rather a sense of being part of an *experience*, most particularly since interactive interfaces can include the participatory action of the audience. The perceptive ambiguity that machines create, accommodates the potential for functional trustworthy design as well as the aesthetic experience, exempted from the necessity of trust. From this perspective, data-driven design does not need to expose its protocol, because the exposition context legitimizes its sole aesthetic experience (even for functional programs).

Finally, the perceived reliability of machines is intrinsically correlated to the trust we attribute to predictability. Consequently, the trustworthiness of design is also entangled with the preconceived predictability of the potential output. Computers do not expose directly their internal functioning, they expose the computed result of the internal execution of a program that is expected, from an external standpoint, to be strictly predictable.

By that I mean that machines are bound to causality, in the Aristotelian sense with a *one cause to one effect* relation. Paradoxically, if machines were easily predictable, we wouldn’t believe that they’re hiding anything, would we ? Even if the internals aren’t exposed, the predictability of the outcome should help the audience and users build trust in their design. I would say that one of the reasons that machines are hard to grasp is not that they are inherently hiding something but because of the perception we have of them. Our initial perception relies mainly on the observed effect. What I’m trying to make explicit is that we apprehend and perceive all machines as what Heinz von Foerster calls non-trivial machines, as he differentiates them from trivial machines that he describes as follows :

> *« A trivial machine is characterized by a one-to-one relationship between its “input” (stimulus, cause) and its “output” (response, effect). This invariable relationship is “the machine.” Since this relationship is determined once and for all, this is a deterministic system; and since an output once observed for a given input will be the same for the same input given later, this is also a predictable system.[5] »*

Now, let us just think about this statement for one moment : are machines stuck in a one-to-one relationship from input to output ? That is actually a really important matter as it pushes us to try to

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

look at different machines in specific technical ways. Clearly, a finite-state machine[6] could, at first, be thought of as a trivial machine based on the predictability of the system. When you put the right amount of money in a vending machine, you hope that your chocolate bar will predictably be distributed to you. The opposite might very well be infuriating, and when it comes to chocolate, trustworthy machines are critical ! This may well be true when considering the machine as a whole, nonetheless, if we look at the intermediary states of machines, matters become more inconclusive. When there is variance in the input-output relation, it indicates that the states of the machine are no longer bound to this one-to-one causal relationship, more specifically in the intermediary states. Let's go back to von Foerster's definition of non-trivial machines :

> *« Non-trivial machines, however, are quite different creatures. Their input-output relationship is not invariant, but is determined by the machine's previous output. In other words, its previous steps determine its present reactions. While these machines are again deterministic systems, for all practical reasons they are unpredictable: an output once observed for a given input will most likely be not the same for the same input given later.[7] »*

Using our chocolate bar example, the machine needs to take into account all the change (most often limited to one currency) you have in your pocket in order to reach the asking price. So, unless you have one coin and the chocolate bar costs exactly that one specific coin, the machine inevitably must implement all intermediary possible combinations of coins in order to reach the price. Only then will you be able to enjoy your treat. This also means that the implementation, in a strict finite number of possible actions, will transition its internal states depending on the previous steps. The internal states will change according to the coin you slip in, until you've reached the asking price. From this perspective, a vending machine isn't a trivial machine but rather a non-trivial one. But, does this mean that it is less predictable ? I don't think so, and I would add to von Foerster's point : machines are still deterministic whether they are non-trivial or trivial machines. Except that the author says that they are unpredictable. We read into this that what he outlined was that predictability was not to be perceived for the machine as a whole but rather for every action it takes into account.

> *« In order to grasp the profound difference between these two kinds of machines it may be helpful to envision "internal states" in these machines. While in the trivial machine only one internal state participates always in its internal operation, in the non-trivial machine it is the shift from one internal state to another that makes it so elusive.[8] »*

I would argue that the fact that this binary division of trivial and non-trivial machines is not so indisputable seems to push the audience to think of them – all machines – as exclusively non-trivial ones : the deterministic aspects of the machine would then remain the same but the unpredictability characteristic would tend to make us, the audience, think of these machines as entirely non-trivial. Paradoxically, this would undermine Kirk's statements about data-driven design, and emphasize the failures of trustworthy design. The elusiveness would be the result of the perceived ambiguity inherent to machines and their predictability. Though, while Kirk's proposal for trustworthiness in the design process is accurate to describe data-driven design in an exposition context, I would argue that it is the responsibility of the designer to construct the means of trustworthiness despite the machines' ambiguity. I would also add that this is exactly where von Foerster's deterministic machines uphold Kirk's trustworthy design : the external observer, the audience, is required to imagine the internal process. The observer has to speculate on the mechanism of the machine in order to assess the trust he/she attributes to the functional machine and its interface. Thomas Fischer and Christiane Herr would reason similarly :

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

> *« Von Foerster approaches the challenge of determining both machines from the perspective of an external observer who, without insight into their inner workings, must construct a mental model of their inner workings – to "whiten" a "black box" in Glanville's terms.[9] »*

I do not believe that the binary separation white/black box encapsulates all the complexity of our perceptions of machines. Our point is that, when it comes to the exposition of data-driven design, the perceived technical ambiguity of machines (both because of their deterministic and predictable characteristics) leads to the exposition of design pieces that bet on legible meaning, and yet, their non-predictability constrains the access to the meaning itself (the understandable and legible protocol). From the audience point of view, if nothing helps bridge the decoupling of cause and effect, then nothing makes the visual effects trustworthy on their own.

2. Didactics of source

The relation to machines is a double bind, conflicting apparent unpredictability and determinism. As I expressed earlier, the context of exhibitions dictates the preconceived ways in which the reception of an art piece will be apprehended. If one were to walk into a room filled with paintings, he/she would presumably not put into question the « implementation » of the artworks, at least not at first. He/she would likely question the representation itself, subject and author, framing and choices. All of this coming from a prior aesthetic experience. Implementation wouldn't even be a word applied in this context, perhaps techniques would be a more accurate description. Since no mediated operation is performed, except the act of experience, (un)predictability and determinism aren't usually tools to gain knowledge from any non-machine based representation. I wouldn't be stating anything new by reminding the reader that aesthetic experiences have been an important part of the philosophical approach to our relationship to art. But the reflection I am trying to undertake here is : how do the exposition methods of data-driven design establish a particular ambiguous relation of knowledge and experience ? And as we have tried to demonstrate, the trustworthiness of design is inherently dependent on the perceived predictability, though the complexity of data-driven design is its non-trivial technical implementation (or the perception of it). But this does not imply that the process, molding an idea into an object, from cause to effect, is fundamentally different when it comes to comparing paintings and data-driven design. Both are technical practices. Both result in a representation (or presentation) with a *display* potential. Nonetheless, the relation the audience has to the implementation of internal protocols is determined by the immediate context : when it comes to complex electronic machines, the co-existence of a possible network of simultaneous contexts make it all the more ambiguous.

Even if the final object is presented in the exhibition hall, it might rely on a pattern of external dependencies, requesting information to third party servers and databases. Or it might not. Only clear mediation can lift the ambiguity. The intercommunicating machines create a less intelligibly perceivable context, and leave the audience facing only the immediate context of the final object, the displayed object in the exhibition hall. This, I believe, tends to lead the audience to a default position of skepticism as a consequence of the (apparent) decoupling of cause and effect. Especially when the premise is a claim to functional code, predictability and completeness.

> *« While the knowledge or form that lies at the heart of the code promises completeness and decidability, the execution of code is often mired in ambiguity, undecidability and incompleteness. This raises many concrete problems in relation to designing code-based interactions. At core, the problematic instability or slippage associated with code concerns the non-coincidence between knowing and doing (or conduct) represented by code.[10] »*

Though I truly understand this argument made by Adrian Mackenzie and Theo Vurdubakis, that I

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

think is accurate when it comes to the code itself, I also believe that the objectives of data-driven design is merely to reconcile *knowing* and *doing*. In other words, I believe that the exhibition context, though maybe ambiguous in essence, is a perfect context for a didactic approach to knowledge. We could just mention the fact that some of the modalities of exhibition involve the use of devices (such as introductory labels, section labels, captions etc.) that are meant to enable mediation between the audience and the pieces, between the non-displayed contextual information and the displayed object. The paradoxical effect of ambiguity is that it leaves the audience skeptical yet open to reflection. For the exact same reasons the audience can distrust the machines (non-predictability, determinism, unable to understand underlying protocols and so forth), it is also clear that this model of perception is one of preconceived beliefs and not knowledge. Therefore, the door is wide open for a mediated approach to knowledge. The question is then, does it fail by design or does trustworthy design reach its mediation goal ?

Let's now turn to an example. In 2017, Lauren McCarthy and Kyle McDonald presented *MWITM (Man / Woman In The Middle)*. They describe the work as follows :

> *« The title is a play on the term from computer security "man in the middle attack", which is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. In this case, we set up a system to MITM attack our own relationship.*[11] *»*

The two artists would set up an experimental protocol of communication. They created an intermediary, a server between their devices, in order to « intercept » all their message exchanges. They would communicate, more specifically exchange text messages, only going through the designated server at the middle of an otherwise end-to-end communication. As the artists expressed it, the concept of this piece is derived from a classical hacker method called *man in the middle attack*, which operates most of the time as a listener that gathers information from within the communication channel. This traditionally happens in covert operations, most obviously. Where this piece commutes the concept, is by exploring the MWITM potential for synchronizing the conversations while simultaneously clearly disclosing the altered protocol. Their scripts were written with the goal to synchronize their discussions, not monitor their conversations. Both artists wrote scripts (respectively MITM and WITM) that they would then implement in the server, that would symbolically « attack » their digital relation. By doing so, they fabricated a third party interference in their relationship : the machine. It would become a integral member of their digital triad, initiating new conversations or altering the messages, swapping some words, adding or removing others.

I chose this project to exemplify the following two complementary aspects : firstly, for the question it raises about the entanglement and boundaries of our relation to machines, exposing to the audience the non-triviality of machines, and consequently, probing machines' trustworthiness. Secondly, for the exhibited methods the artists decided to expose in order to reveal the machines' technical implications, and consequently, the exposition of the didactics revolving around the conception protocol. Up until now, I have considered ambiguity as an inherent form of material trustworthiness paradox. I would now like to use this case study to take a closer look at how this ambiguity can be extended to both entanglement and boundaries, within the context of exposing machines.

> *« A major implication of entanglement is that boundaries of all kinds have become permeable to the supposed other. Code permeates language and is permeated by it; electronic text permeates print; computational processes permeate biological organisms; intelligent machines permeate flesh.*[12] *»*

Katherine Hayles stresses the importance of discerning that permeability is an inherent

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

characteristic property of entanglement : what I have stated as the ambiguity of the hardware/software relation, she theorizes that it is the key structure for understanding the complex dynamics of reciprocal intermediation. Mackenzie's and Vurdubakis's instability or slippage comes from the separation at boundaries, which Hayles refutes, stating that permeability does not imply that no distinction exists. Not only this, Hayles also calls for this approach to be the one guiding our awareness of inter-penetrable boundaries. In our example, the interception and manipulation of information plays this exact role ; while it breaks down the intersubjective end-to-end communication, it also adds a layer of technical intermediation. The entanglement is materialized by the server synchronizing the exchanges. It showcases the permeability of boundaries from a technical standpoint (the « attack » itself) and from a social standpoint (the entanglement of technologies in conversations). The artists also accentuate the reciprocity of intermediation by updating their respective scripts (MITM and WITM) in order to achieve a « better relationship and conversation[13] », eventually aiming for a fluid boundary-less human-machine-human intermediated relation.

> *« Rather than attempt to police these boundaries, we should strive to understand the materially specific ways in which flows across borders create complex dynamics of intermediation. At the same time, boundaries have not been rendered unimportant or nonexistent by the traffic across them. Biological organisms are not only computational processes; natural language is not code; and fleshly creatures are composed of embodiments that differ qualitatively from artificial life forms. Boundaries are both permeable and meaningful; humans are distinct from intelligent machines even while the two are become increasingly entwined.[14] »*

In their attempt to synchronize their relationship, McCarthy and McDonald chose to exhibit this piece by displaying it as a quadriptych. At the center, both smartphone devices show a snippet of the conversations, which sets up a direct comparison for the audience. Therefore, a spectator can notice the subtle changes in the end user experience, either initiated by the server (a message only appearing on one device) or a discrepancy in the exchange (the addition of word, of a smiley face to a message and so forth). On either side of the two devices, the audience can read a print-out of an undisclosed version of the respective scripts. Exposing the source code is rare enough in the art world to be noted. Or, to be more precise, it is rarely exploited as a display piece. When it is exposed, it is usually part of an open source policy, a sort of footnote to a project not relating to the main discourse of the piece, and most often then not accessible only online. I believe that this aspect somewhat shifts the non triviality of a project. A didactic approach of exposing source (as an integral part of the final displayed piece) readies potential legibility and gives access to all potential internal states of the server. Does it mean that this piece achieves Kirk's goal of trustworthy design ? Perhaps not directly. At least, it provides the means of legibility by lifting the veil of the technical implementations. It doesn't mediate the source itself and doesn't try to remove ambiguity ; it exposes the entanglement by exhibiting a didactic framework for understanding the piece. By doing so, it seems as if it removes distrust as a preconceived perceptive apprehension. Even if this specific piece doesn't rely on live data feed, it does rely on a network of technical contexts and sets the didactics of source as a design strategy for exposing potential representations of information.

3. Strategic fail-safes

Up to this point, we've seen that : first, trustworthiness is a key component on both the way we perceive an exhibited design object and the design process, and secondly, that even though the inherent technical entanglement lends itself to distrust in the way we perceive machines, the context (or multiple thereof) of exhibits provides a potential for a didactic approach to knowledge, for intermediation. Going forward, we will take a closer look at some of the strategies put into place, to remove or to emphasize the ambiguity. Now, some of the ambiguity doesn't necessarily directly depend on the piece itself. It is rather a matter of trying to alleviate some of the puzzles of

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

data-driven design's technical implications and ramifications.

Let's initiate this point in our discussion by mentioning a fact : data-driven design is intrinsically compelled to take into account the potential failure of both technology and the context-specific issue that could arise. In an exhibition, this can be as simple as making sure that a piece that depends on the internet to gather its live data, actually does have access to the data source. For all practical purposes, this is in effect both a question of design itself (embedding solutions at the piece's core) and of exhibit management (guaranteeing the access to a network). Unfortunately, even with the greatest determination and good faith, exhibitions can't provide a perfectly foolproof environment for hosting data-driven design. Failures happen. The fact that a failure is quite predictable signifies that strategies can be set up to mitigate potential future problems with fail-safes. I have to underscore that *fail-safe* doesn't mean *fail-proof*, only that the design itself provides multiple paths in case of failure, in order to prevent worst case scenarios with more suitable ones. In an exhibition, the possible outcomes are probably not going to be disastrous. But fail-safes are extremely important in all sorts of systems, going from spacecraft or voltage regulators to kitchen sink drains. In order to identify which scenarios are to be expected in a specific context, one has to take a look at the transitions of internal states within the machine.

To express it simply, these internal states are like crossroads that dictate potential paths. The end goal is for a displayed piece to have a consistent outcome whenever a failure situation arises. Fail-safes implement predictable outcomes with non-trivial implementations of internal state transitions. As a whole, the piece needs to be functional and trustworthy. As a fragmented layering of internal states, it needs to be adaptive and entangled with its context. And this is where the ambiguity comes in handy : nothing prevents creators from displaying a piece that switches from live data feed to backup, without explicitly giving that information to the audience, without exposing itself. As we'll see with Samuel Bianchini's example, interdependence with third party services can be quite tricky to handle.

In 2009, Bianchini produced *AllOver*, an online artwork presented both on the internet and in several public exhibitions over the years. The piece transforms still photographs into ASCII art[15] by changing them according to the volume of transactions based on financial data. It is described as follows : « the figures and letters composing the images are dynamic and keep changing: they are generated in real time following the rise and fall of stock market indexes around the world.[16] » The sentence makes it quite explicit that the piece is based on *real time* data. The spectator, at first, will doubtless apprehend the work believing that what he/she sees as a whole is live data-driven artwork. Except that several other clues tend to make us reassess this initial statement. In the caption, it is mentioned that : « Part of Data provided for free by IEX. View IEX's Terms of Use.[17] » *Part of the data* ? Where does the *rest* of the data come from ?

Now, if we look at the generated image itself, at the very bottom in the caption, one can read two different timestamps. One being the *present* time and date, which tends to corroborate the *real time* claim of the piece. However, a second one is sometimes visible, displaying a timestamp prefixed by : « Data recorded on mm.dd.yyyy[18] ». This secondary statement seems to contradict directly with the perceived real time data-driven interface. So, to sum it up, we have two parallel assertions : the first, a definite descriptive affirmation of *real time*, and the second, a subdued clue for some sort of technical switch from one data source to another. These opposite and simultaneous discoursive items of information are most clearly ambiguous.

I took a dive into what I could technically reverse engineer solely through the reading of the code of the public internet version. What I could unravel is that the front-end script uses a fail-safe backup system : it tests its access to the third party data provider (IEX) and if the test fails (data source offline or no internet connection), the data is retrieved from a locally stored backup file. As of the writing of this paper, the latest version of this backup file[19] ranges from 2017-10-18 17:46:26 to 2017-10-31 10:42:41, and contains 20000 entries. Let's look back at the two apparent opposing arguments : is there a change of internal states to switch data source ? Is it real time ? First, the answer seems to be yes. Despite potential failures, a fail-safe system enables the data-driven art piece to continue its functioning independently of external and contextual factors. Secondly, one

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

could argue that *real time* is only a matter of *real time data analysis* and not *real time data*. While this is true, it seems once again to leave it up to the audience to discern the technical subtleties and to identify the use of strategic ambiguity to dissimulate the fail-safe mechanism. What this indicates is that, while the real time statement is in effect still valid, the ambiguity revolving around the source of the data remains. As I mentioned earlier, I am not trying to say that this ambiguity is intended to misdirect the spectator. Interpretation of the artist's intention is not my point. But by trying to implement fail-safes and not clearly lifting the ambiguity regarding its operation, the artist chooses to leave doubt concerning how his work functions internally. The complexity of this endeavor resides in the fact that potential failures are *not* in the artist's control, but stem from a network of systems to take into account : access to the internet in an exhibition hall is handled by the exhibition management itself, access to the data is defined by the third party terms of use and most often than not have strict restrictions including financial ones. In short, three matters are taken into account : exhibition specificity, technical implementations, and financial constraints.

It also appears that the more a piece is entwined in a network of technical systems, setting up potential strategies can become more and more laborious. The interdependence of services delegates the survival of a piece to external parties. If a failure arises, then the whole network is put into jeopardy. In their very well documented 2015 installation entitled *Artificial Killing Machine*[20], Jonathan Fletcher Moore and Fabio Piparo built a live data-fed machine that activated itself every time a U.S. drone strike occurs.

> *« This time based work accesses a public database on U.S. military drone strikes. When a drone strike occurs, the machine activates, and fires a children's toy cap gun for every death that results. The raw information used by the installation is then printed. The materialized data is allowed to accumulate in perpetuity or until the life cycle of either the database or machine ends.*[21] *»*

While Bianchini tries to avoid the hazards of external dependencies by implementing a backup system, Piparo and Fletcher Moore explicitly anticipate failure as an integral part of the data-driven installation. Opting for a didactic approach to the internal functioning mechanisms, the artists also propose a deathbound data-driven piece. It will fail, with no fail-safe strategies in place, offering instead a simple disclaimer of the delegation of control over the survival of their installation. In its functional state, this is how it operates : « The application queries this Internet database every five minutes and when *a entry been* [sic] detected in the database, the motor control functions activate.[22] » This artwork relies on live data access through the use a public API[23], *dronestre.am*[24], provided by another data artist, Josh Begley. His dronestream API (and therefore, indirectly the *Artificial Killing Machine*) exploited the data from a journalistic source called : *The Bureau of Investigative Journalism*[25]. What this source provides is a plain flat file spreadsheet of U.S. drone strikes, and what Begley created was a technical bridge between a human legible database and a computer exploitable interface. Dronestream was the intermediation of data and actuation of the machine : it enabled it to function. When the *Artificial Killing Machine* was exhibited in 2015, the API had been publicly available for a couple of years. On November 16th 2017, Josh Begley published a tweet on the @dronestream account : « After 5 years, I think @dronestream is over.[26] » As of that point, all live access to the dronestream API was no longer reliable. Most of the data is still accessible but mostly useful as a historical database of U.S. drone strikes (covering a period from early 2002 to early 2017), not as live updated feed[27]. Paradoxically, at the time of writing of this paper, the source data from *The Bureau of Investigative Journalism* is still being updated. By examining and comparing the data sources, it seems as if the last update of the API was in early 2017[28]. This means that we can presume that as of 2017, the *Artificial Killing Machine* is no longer a functional machine because of the deprecated API. Does this signify that the artwork is dead ? This is our case in point : the piece integrates this failure, it is embedded within the concept of the artwork itself. It predicates the potential (non-trivial) outcomes because of its dependence on external services that might collapse.

Both strategies we have examined attempt to expose and alleviate some of the particularities of

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

data-driven design in contrasting implementations. The fact that, as suggested by Drucker, aesthetics is transformed with intermediation at the inherent core of system-based relations, leads us to perceive these occurrences of strategic ambiguity as efforts to reconcile *use* and *discourse*. These efforts are all the more crucial, knowing that the context of exposition presents itself as the focal point for perceiving complex networks of interdependence. While looking at a painting, it seems as if the caption describing the author's techniques extends the observer's contextual knowledge. As for data-driven design, exposing these technical mechanisms generates new tools of knowledge for the audience, and shows that these technical implementations are an integral part of the author's choice of creative strategies.

4. Limits

The trustworthiness that we, as an audience, perceive and attribute to machines is bound to its design, both through the creators' code-based choices and the inherent technical aspects of system-based devices. As we have seen, some creators embrace the machine's ambiguity by offering a didactic and mediated approach to data-driven devices. Others set up strategies to bypass the potential failures and ramifications specific to the exhibition context. But what are the technical limits to our understanding of technical design ?

Even willingly deconstructing a technical device is most of the time restricted, and maybe won't explain anything about its functioning. *Predictive Art Bot*[29] is an artwork by Disnovation. By being presented in both the context of an art installation (notably at the Centre Pompidou's *Coder le Monde* in 2018 and at the ZKM's *Open Codes* in 2018) and as an online project, it made it possible for me to dig into its internal functioning. Or at least try. As with the previous examples, all the methods I use are extremely simple and don't require any form of hacking, which I believe would break the purpose of my experiments. They are merely the result of my attempt to read design and art by setting aside the top layer, if only for a few minutes. As you will see, you don't learn much and that is my point. In the exposition context, the only thing one can do is trust, by default, the descriptive texts accompanying the piece : « An algorithm that turns the latest media headlines into artistic concepts. » The explanation also mentions the fact that it intends to caricature « the predictability of media influenced artistic concepts. » Finally it states the following about the mechanisms : « it identifies and combines keywords to generate concepts of artworks in a fully automated way[30] ». From what I could unravel, the client side requests a json encoded feed from a websocket server, every 20 seconds. The data source contains a list of different sentences[31]. In laymen terms, this means that without having recourse to more complex methods, there is no way of knowing if the sentences are actually generated by an algorithm or just randomly chosen from a prebuilt database. One has either to trust such a piece that offers no insight on the entwined ambiguous mechanisms, or to resolve oneself to distrusting the piece because of its failure to expose its structural meaning. Again, the mediated information is what enables the audience to construct its own experience. Multiple injunctions lead in different, polarized, perceptive directions : the use of the word « caricature » would legitimize a speculative approach to design, but the clear mention of phrasing such as « continually monitors emerging trends » can exacerbate the piece's ambiguity. It is fair to ask ourselves if these data-driven pieces really produce meaning : « A basic distinction can be made between visualizations that are *representations* of information already known and those that are *knowledge generators* capable of creating new information through their use.[32] » Applying this distinction made by Johanna Drucker, it seems clear that the more data-driven design structures itself around the exploitation of non-trivial paradigms (non-predictable determinism), the farther the gap deepens between the audience's perception and the generated knowledge. This is especially noteworthy if the claim is founded on an algorithm generating novelty, even if this novelty accommodates a critical discourse. I think that this is visible particularly in the contexts of exhibitions but reveals a broader, paradoxical, human-machine social relationship. « Computers have fostered both a decline in and frenzy of visual knowledge. Opaque yet transparent, incomprehensible yet logical, they reveal that the less we know the more we show (or are shown).[33] » So where does this leave us in regard to our initial questioning about legible meaning in the context of exhibiting/exposing software ?

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

I will conclude by saying that the existing protocol guiding our relation to visual knowledge is slightly outdated (if I may say, for lack of better terminology), and, most importantly, profoundly divided. The entanglement of knowledge and (inter)action shifts the perception of software, mainly because of the transformation of aesthetic experience identified by Drucker. The proliferation of data visualizations, outside the experimental realm of art and design exhibition, has made users more and more accustomed to both representations and knowledge generators. But being accustomed does not intrinsically make the design trustworthy. Nor does it demonstrate its capacity to generate knowledge. Making it technically possible for a network of different simultaneous contexts to be entwined generates a whole new level of ambiguity that creators can embrace or try to evade. By choosing a didactic mediated approach and exposing the sources for both data and code, the audience is trusted with a legible unveiling of the machines internal states. And by trying to bypass the complex technical constraints of the simultaneous contexts (from the servers, API, to the context of the final piece), the audience is left with a functional system-based object where the predictability of the whole piece overshadows the meaning of the piece itself. As we tried to read into the technical aspects of design, it becomes clear that none of these aforementioned strategies can lift all the limits to the acquisition of knowledge, at least, from an audience's perspective. Nor do they have to : data-driven design is ambiguous and can swing from representation to knowledge generators, and vice-versa. This, I believe is what makes exposing software at the same time so fascinating and scary, like giving away the method of a magic trick. Exposing the causes doesn't change the effects, it merely includes the audience in the entangled complexity of human-machine relations and entrusts it, for it has the potential of attain new layers of knowledge and not simply experiencing visual representations of displayed design.

Bibliography

Chun, Wendy Hui Kyong, *Programmed Visions: Software and Memory*, Cambridge, MA, The MIT Press, 2011.

Drucker, Johanna, *Graphesis : Visual Forms of Knowledge Production*, Cambridge Massachusetts USA, Harvard University Press, 2014.

Drucker, Johanna, *SpecLab : Digital Aesthetics and Projects in Speculative Computing*, Chicago, University of Chicago Press, 2009, p. 177.

Fischer, Thomas, Herr, Christiane M., « An Introduction to Design Cybernetics », Fischer, Thomas, Herr, Christiane M., dir., *Design Cybernetics : Navigating the New*, Cham CH, Springer Nature, 2019, pp. 1-23.

Galloway, Alexander, *Protocol: How Control Exists After Decentralization*, Cambridge, MA: MIT Press, 2004.

Hayles, Katherine, *My Mother Was a Computer: Digital Subjects and Literary Texts*, Chicago, University of Chicago Press, 2005.

Kirk, Andy, *Data Visualisation : A Handbook For Data Driven Design (2nd edition)*, SAGE Publications Ltd, London, 2019.

MacKenzie, Adrian, Vurdubakis, Theo, « Codes and Codings in Crisis: Signification, Performativity and Excess » in *Theory Culture Society*, vol. 28, n° 3, 2011, pp. 3-23, DOI: 10.1177/0263276411424761.

von Foerster, Heinz, *Understanding Understanding : Essays on Cybernetics and Cognition*, New-York, Springer-Verlag, 2003.

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

1. Drucker, Johanna, *SpecLab : Digital Aesthetics and Projects in Speculative Computing*, Chicago, University of Chicago Press, 2009, p. 177.
2. Galloway, Alexander, *Protocol: How Control Exists After Decentralization*, Cambridge, MA, MIT Press, 2004, p. 7 and p. 243.
3. Kirk, Andy, *Data Visualisation : A Handbook For Data Driven Design (2nd edition)*, SAGE Publications Ltd, London, 2019, p. 15.
4. *Ibid.*, p. 227.
5. von Foerster, Heinz, *Understanding Understanding : Essays on Cybernetics and Cognition*, New-York, Springer-Verlag, 2003, p. 208.
6. A finite-state machine or automaton is a machine that contains only a defined number of states and that can only be at one state at any specific time. The given state is one of the finite possible states of the machine. Many machines like vending machines are finite states machines.
7. von Foerster, Heinz, *op. cit*.
8. *Ibid*.
9. Fischer, Thomas, Herr, Christiane M., « An Introduction to Design Cybernetics », Fischer, Thomas, Herr, Christiane M., dir., *Design Cybernetics : Navigating the New*, Cham CH, Springer Nature, 2019, pp. 1-23, p. 13
10. MacKenzie, Adrian, Vurdubakis, Theo, « Codes and Codings in Crisis: Signification, Performativity and Excess » in *Theory Culture Society*, vol. 28, n° 3, 2011, pp. 3-23, DOI: 10.1177/0263276411424761.
11. https://lauren-mccarthy.com/MWITM-Man-Woman-In-The-Middle, [visited on September 26th 2020]
12. Hayles, Katherine, *My Mother Was a Computer: Digital Subjects and Literary Texts*, Chicago, University of Chicago Press, 2005, p. 242.
13. https://lauren-mccarthy.com/MWITM-Man-Woman-In-The-Middle, [visited on September 26th 2020]
14. Hayles, Katherine, *op. cit*.
15. ASCII art is a technique consisting of using ASCII encoded characters to form an image.
16. https://dispotheque.org/en/all-over, [visited on September 28th 2020]
17. Ibid.
18. For example : « Data recorded on 10.23.2017 | 10.06.2020 12.11 PM | FRA: DAIO PAPER CORP. »
19. https://allover.dispotheque.org/backup.json, [visited on September 28th 2020]
20. Artificial Killing Machine, https://www.polygonfuture.com/artificial-killing-machine, *[visited on September 24th 2020]*
21. Ibid.
22. Ibid.
23. Application Programming Interfaces are technical interfaces designed for multiple software to communicate with each other. They are at the core of interdependence of services.
24. http://dronestre.am/, *[visited on September 18th 2020]*
25. https://www.thebureauinvestigates.com/stories/2017-01-01/drone-wars-the-full-data, *[visited on September 25th 2020]*

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5

26. https://twitter.com/dronestream/status/931262631381471232, [visited on September 18th 2020]
27. Though http://api.dronestre.am/ is totally offline, the direct access to http://api.dronestre.am/data is still online, [visited on September 28th 2020]
28. The latest entry is YEM262, date 2017-03-06T00:00:00.000Z in the API and 30/01/2017 in the *The Bureau of Investigative Journalism*'s database. This discrepancy might indicate that the actual API was not actually real time strikes, but rather a delayed updated database based on the when the event was undisclosed.
29. Available online http://predictiveartbot.com/ [visited on September 23rd 2020]
30. https://disnovation.org/pab.php, [visited on September 23rd 2020]
31. Data structure : {type : "sentence", data : { sentence : "xxx", articles : […], all_articles : [...]}, id : xxxxxxxxxx }
32. Drucker, Johanna, *Graphesis : Visual Forms of Knowledge Production*, Cambridge Massachusetts USA, Harvard University Press, 2014, p. 65.
33. Chun, Wendy Hui Kyong, *Programmed Visions: Software and Memory*, Cambridge, MA, The MIT Press, 2011, p. 16.

© La revue DAM
téléchargé le 2026-11-21 03:12:28, depuis le 216.73.216.5