Code and Material

Georg Trogemann

This is the English version of the introduction to »Exkursionen ins Undingliche«, in: Georg Trogemann (Hrsg.), Springer Wien/New York, 2010, 15-26.

"We do not live in the age of materialism, as all the philistines complain, but rather in the second Platonic age. Only today, in the epoch of mass industry, is the individual object actually assigned a lower ranking than its 'idea'," Günther Anders wrote in 1978, on "The Antiquiarianess of Materialism."[1] It has, indeed, been a long time since owning the "non-physical recipe" for the production of innumerable copies of a thing became more economically relevant than owning the thing itself. And this knowledge about things is now no longer stored in the brains of craftsmen or recorded in writing, but rather as digital information, as an un-thing (*Unding*: Flusser). With this we enter the world of algorithms. Today we program machines so that they spew out things, which have become worthless, en masse. We design new things on computers and develop control software that makes it possible for products to be produced in endless variations. No car is exactly the same as the next, no IKEA kitchen just like that of the neighbours. Programed pseudo-individuality is one of the hallmarks of the digital industries.

In 1984, Jean-Francois Lyotard tried to examine the change in our perception of the materiality of things through the development of digital information systems with his Paris exhibition *Les Immatériaux*.[2] The exhibition "[...] touched on interests that extend deep into everyday life and popular culture. With the development of media technology, along with the much-discussed notion of the extension of the body and ubiquitous surveillance, the world seems to forfeit its material differences and the perception of fleeting surfaces takes the place once held by solid objects. The Parisian exhibition was, however, received less as a critical interrogation of the suitability of the traditional concept of material, and more as confirmation of the overcoming of old world physical materials through information technology's supposedly immaterial images. With this, digital codes are bestowed a status that was once only attached to artistic work, as the transformation of material into another, higher state."[3] The Paris exhibition's reception reflected the widespread notion that there are high and low forms of culture. The high forms include the intellectual, immaterial forms of expression; material expression belongs to the low forms. This hierarchisation of the cerebral and the material runs through the whole history of art in the West. Concept pairs such as *code and material*, *symbol and thing* are, from this perspective, merely modern variants of an old duality. In the hierarchisation of thing and symbol, the symbol currently holds the upper hand

[1] Günther Anders, "Die Antiquiertheit des Materialismus," in *Die Antiquiertheit des Menschen*, 3rd ed. (Munich: C.H.Beck, 2002), 2:37.
[2] See also Antonia Wunderlich, *Der Philosoph im Museum* (Bielefeld: transcript, 2008).
[3] Monika Wagner, "Material," in *Ästhetische Grundbegriffe: Historisches Wörterbuch in sieben Bänden*, eds. Karlheinz Barck et al. (Stuttgart/Weimar: J.B. Metzler, 2001), 3:867.

once again. “This is the transition from an ‘ontological symbolism’ to an ‘operative symbolism’. ‘Ontological symbolism’ represents the view that the givenness of symbolically represented objects is independent from the act of symbolisation, that objects precede ‘their’ symbols, and the symbol holds a merely secondary status. The term ‘operative symbolism’ accentuates conversely the conviction that sensorially perceivable symbols are primary compared with what they represent, that the objects symbolised are first created through this act of symbolic reference. The transition from ontological to operative symbolism can be emphasised thus: no longer do things lend symbols their meaning, rather it is symbols that constitute things as epistemic objects in the first place.”[4]

The Materiality of Information

Let us take the example of a simple object, a circle. The separation of the idea of a circle from its material realisation is easy to comprehend. From early cybernetics: “When we draw a circle, the geometric informational content that this shape represents is completely independent of the materiality that comes into play in drawing it. We may draw the circle on a blackboard with chalk or with a pencil on paper. This change of materiality, in which the drawing manifests, is just as irrelevant as the materiality of the cybernetic system (the person drawing) that produces the shape. If it is also asserted that an information system as such is totally independent from a given material existence of the world, then *matter* is always meant in the sense of wood, chalk or also flesh and blood. It is precisely in this that the revolutionary meaning of cybernetics lies: the assertion that properties and behaviour, which we have attributed exclusively to living flesh and blood in the past, can also be realised independently of such specific materiality.”[5] Today we experience not only an immense increase in ideas, an “inflation of inventions, which don’t appear anywhere as ‘production of ideas’ in Plato,”[6] but also in particular a change in the representation of ideas. The idea of the circle now coincides completely with its mathematic formalisation. In the ideal case, ideas exist as formal procedures, i.e. as algorithms. The step to the production of concrete instances of the idea is then a very short one.

Abraham Moles recognised the importance of information as an artistic material very early on. “An awareness of the materiality of information is extraordinarily young. [...] first with the invention of other communication channels [after the book], with the telephone, radio, recording of sound, image and movement—homo faber was once again a step ahead of homo sapiens—did one notice that this materiality went beyond the weight of the paper and the number of telephone cables, it was, namely the symbol; one understood the existence of the materiality of any communication. The *ideality* of communication was thus reduced in its importance in conceptual representation through the addition of materiality. [...] From now on there is a whole category of individuals in the modern world who deal with material bearers of ideas: no longer printers, booksellers, messengers and switchboard operators, but communication engineers. For them, the signal carrying ideas that they do not know and don’t have to take care of goes through telephone wires, cables and speakers; they must, however, deal with problems of overloaded wave lengths, the capacity of telephone networks or even more tangibly with telegrams fees; for them the quantitative aspect of information is obvious. [...] Modern technologies have created, successively, the page-setter, which was once an insignificant role in the printing business; the cutter, who is often the real film director; and

---

[4] Sybille Krämer, “Kalküle der Repräsentation,” in *Räume des Wissens: Repräsentation, Codierung, Spur*, eds. Hans-Jörg Rheinberger et al. (Berlin: Akademie-Verlag, 1997), 111.

[5] Ross Ashby, as quoted by Gotthard Günther, in *Das Bewusstsein der Maschinen* (Baden-Baden: Agis Verlag, 1963), 117.

[6] Anders, “Antiquiertheit.”

the audio engineer in broadcasting or the recording studio, and placed them in these positions as artists. [...]. A new aesthetic must arise on this basis; initially it systematically examines the materiality of communication and not its ideality, which had constituted the subject of classic aesthetics [...]."[7]

So, the concept of information clearly does not manage to overcome the old dualism of ideality and materiality either. As the word "in-formation" already suggests, it is primarily about the *form in things*.[8] Consequently, information also needs a material that can take on this form. Of course, every material that we form is always already the result of an objectification itself. In order to be able to use and revise a material purposefully, we must be able to explicitly name its properties and its behaviour under various conditions. This means that the mere perception of the material is not enough—we need at the very least a differentiated notion, a plan as to how we can handle and change the material for our purpose. Richard Sennett talks in this context of *material consciousness*, which not only the craftsman, but also anyone who works with symbols, must have. "Is our consciousness of things independent of things themselves? Are we aware of words in the way we feel an intestine by touch? Rather than get lost in this philosophical forest, it might be better to focus on what makes an object interesting. This is the craftsman's proper conscious domain; all his or her efforts to do good quality work depend on curiosity about the material."[9] Here we finally come to the questions of praxis. With Richard Sennett the contest between thing and consciousness can safely be left to the philosophers; it is more fruitful for artistic practice to not see material as the antithesis of thinking and reflection.

Not only for the craftsman's work, but also for the artistic works brought together here, it is more useful to understand the material and immaterial as belonging together, as something simultaneous, and mutually dependent. Thought does not precede material things and is thereby not its actual origin, nor is the immaterial subordinate to the material and completely determined by it. With Merleau-Ponty we can say: "There is not only pure consciousness on one side, things on the other. We are particularly interested in the interstitial realm between consciousness and things." The somewhat cumbersome combination *code and material* attempts to broach precisely the issue of this realm with a special focus on the computer and digital media. Neither natural scientists nor visual artists, especially those who produce programed artefacts, can get by without an awareness of material. Anyone who builds experimental *self-active objects* is exposed to the strange mixture of code and material. These are objects that are embedded in a chosen environment and act there under the control of a program. Such *algorithmic artefacts* are not aimed at overcoming materiality, but rather at updating a traditional concept of material.

The coding of thought

Computer codes are strange hybrid objects. The symbols that operate in computers are rooted both in mathematics and electrical engineering. They are, on one hand, text and thus to be read as mathematic-symbolic descriptions, on the other hand they possess, as material signalling processes in the computer, the power to translate symbolic operations, thought out by programmers, into real actions.

---

[7] Abraham A. Moles, *Informationstheorie und ästhetische Wahrnehmung* (Cologne: DuMont Schauberg, 1971), 254-256.

[8] See Georg Trogemann, „Algorithmen im Alltag", in: Exkusionen ins Undingliche, Georg Trogemann (Hrsg.), Springer Wien/New York, 2010, 159-185.

[9] Richard Sennett, *The Craftsman*, (Connecticut: Yale University Press, 2008), 120.

Due to its materiality, the machine's hardware is "bound to nature," it is subject to the laws of our physical work and is embedded in it. In order to do its work it uses energy, it is subject to the processes of aging and is prone to all kinds of defects. At the same time it is, however, capable of engaging in the world via connected actuators. Codes, on the other hand, are directly related to human thought and human language, they are bound to algorithmic rationality. Codes do not abide by physical laws, but logical ones. This is the art of the designer. They bring material in such a form that something new emerges, the qualities of which can no longer be described on the material level. In the case of the computer, formal computational models are realised on silicon. Here we are subject to the boundaries of the operational use of symbols, physics takes a step into the shadows. The essential characteristic of the computer is therefore not digitality, but rather its formal-logical foundation. Even the fact that we can effortlessly describe and deal with constant processes and systems on a symbolic basis, demonstrates digitality's lack of importance. Errors that arise in connection with program codes are therefore always errors of reasoning and never material errors in a physical sense. Errors in reasoning arise if a programmer's expectations of their code do not correspond with the functions the code actually computes. The strict division between *interpretation* in the semiotic sense and *execution* in the sense of computation is decisive in this interplay of programmer and computer. It is always the programmer that *thinks ahead* and captures their thoughts in symbols, and it is the computer that *computes after*. It has no possibility to interpret the code, given as a sequence of steps, it can only execute it. If the machine also outputs the results of the computation as symbols, it is in turn the human user that interprets these computed symbols, not the machine.

We can understand program codes as disembodied chains of action that must imprint themselves in a material, via the intermediate stage of a symbolic system, in order to be able to take their effect.[10] A translation process takes place in the course of programing; thought processes are transferred into active mechanical processes. The formalisation, i.e. the applicability independent of concrete insight, plays an important connective role in this. Formal methodology attempts to create unreflective repeatability, a basis of prerequisites that is always in play, but does not always need to be updated.[11] Codes describe objectified knowledge that becomes operative through the execution of individual instructions. Programs are a form of theory of action, which Holling and Kempin term *implemented theory*. [12] An *implementation* is to be understood as the execution of a formal process—an algorithm—in a machine. There the process can then run, completely detached from the programmer. In this image, codes are the joints that connect human thought with mechanical action.

The formalisms of mathematics and the natural sciences have reached new heights in the program codes of digital machines. Formalisation means major abstraction from real circumstances, so a withdrawal, generalisation and cleansing of the insignificant and ambiguous. The different modalities of media dissolve almost completely on the formal level of the code and find a common basis. The conventional division between image, sound, movement etc. is now only intact on the surface; structural equality between image and sound has long been established as a basis on which to work in the theories that are responsible for the layers underneath, for example in signal processing. In the computer, clear and selective formalisms now carry out operations and create new phenomena. The fact that the abstraction process runs backwards in the processing codes is decisive in this. The interfaces load the

[10] See Georg Trogemann, „Algorithmen im Alltag", in: Exkusionen ins Undingliche, Georg Trogemann (Hrsg.), Springer Wien/New York, 2010, 159-185
[11] See Hans Blumenberg, *Wirklichkeiten in denen wir leben* (Stuttgart: Reclam Verlag, 1999).
[12] Eggert Holling and Peter Kempin, *Identität, Geist und Maschine: Auf dem Weg zur technologischen Zivilisation* (Hamburg: Rowohlt Taschenbuch Verlag, 1989).

results of the computation with ambiguities and vagueness. But it is not that which was removed in the process of abstraction and formalisation that is now added back—the gaps are filled with the structures of the machine. The underside of the media—the code—remains strictly rational, but on the surface—the interface—the phenomena are now augmented by unintended artefacts and side effects of the symbolic operation, opening up new interpretational scope for the user.[13] In this way, form of reason and aesthetic form become inseparably linked.

The materiality of the code on the idea-bearer *surface*, which Abraham Moles describes for printing, also plays a role in code development. The question here is to what extent the written code supports the mechanisational thinking of the programmer. The development of various programing methodologies—from *instructions* via *functions* to *procedural*, *logical* and *object-oriented* programing languages—has a major role in the legibility or comprehensibility of the program text and the support that this can already afford through the formatting, layout and allocation of symbols and text blocks on the material carrier of ideas. In modern software tools, the development process is supported visually as well as functionally in a number of ways. In a cooperative process, the tools are co-authors of the code. The powers of convention and habit in programing should also not be underestimated. "The 'gradual fabrication of thoughts' in the production of software is bound to the linguistic devices that are available. [...] In practice the selection of programing language is only rarely based on rational decision; tradition, habit, myth all insist on playing a part. Style refers—in art as in the artificial world of formal artefacts—to the cultural context in which it was formed."[14]

In programing, at the level of the code, we do not only make rational decisions but also always aesthetic ones. How thick should the line of the circle be, what colours are chosen for the figure and the background, how big should the circle be in relation to the total area, which resolution is to be chosen? However, these aesthetic decisions in the code will have to be made with consideration for the materiality of the realisation. And even the algorithmic-rational decisions within the chosen algorithm must address the materiality of the computer, inasmuch as disk space and temporal behaviour will possibly influence the choice. Codes are thus a composite that merge immaterial algorithms with material interfaces to form something like an *informed material*.

Code as composite

When we talk here of code as material, we mean an immediate concept of material that addresses the objectifiable side of the world of things. The two crucial characteristics of every materiality are *measurability* and *changeability*. *Measurement* registers quantitative-qualitative distinctions and thus makes it possible to establish the individuality of things in the world. *Changeability* means that we can shape material and make it serve our purposes. The designer Gui Bonsiepe uses the Heideggerian concepts of *Vorhandenheit* (presence-at-hand) and *Zuhandenheit* (readiness-at-hand) in order to characterise design processes in general as activity that transforms the merely *present-at-hand* into something *ready-to-hand*.[15] "For this purpose 'present-at-hand' (raw, disorderly, incommunicable) materials are shaped and

[13] On the double existence of media contents and the difference between "surface" and "under-surface" see Frieder Nake, "Das doppelte Bild," in *Bildwelten des Wissens: Digitale Form*, ed. Margarete Pratschke (Berlin: Akademie-Verlag, 2006), 40-50.

[14] Jörg Pflüger, "Über die Verschiedenheit des maschinellen Sprachbaus," in *Computer als Medium*, eds. Norbert Bolz et al. (Munich: Wilhelm Fink Verlag, 1994), 161.

[15] Gui Bonsiepe, *Design - the blind spot of theory or Visuality | Discursivity or Theory - the blind spot of design: Conference text for a semi-public event of the Jan van Eyck Academy, Maastricht, April 21, 1997.*

structured in a design and production process, in order to transform them into a 'ready-at-hand' (usable, structured, need-oriented) artefact. The material of a design can be both physical-material as well as of symbolic nature. The readiness-at-hand of an artefact is measured by the level to which it enables efficient handling, whereby design, alongside physical and cognitive efficiency, is particularly aimed at social and cultural efficiency. A characteristic of design is therefore that social and cultural contexts are interpreted in them. The artefact as the result of design thereby transports this interpretation into the world, that it offers a structured scope of action in regards to its use."[16]

Just as, for example, fundamental things can be said about the material properties of wood for organ building—most of which applies to the use of the wood as a raw material in general—there are also fundamental things that can be said about computer codes as the material of artistic works. The material sciences are not interested primarily in the physical composition of materials, rather in particular in the processing and functional properties. A material that has excellent properties in the production of an artefact, must not necessarily have the same qualities with regard to longevity and maintenance and vice versa. How the material can be worked, right down to questions of availability and price, is particularly important in the manufacturing process. Questions of ageing and maintenance first play a role in its use and its lifespan. The materiality of carriers of ideas, which Abraham Moles talks about in the example above, also has relevance for the production of codes. Moles explains the importance of the materiality of idea-carriers using the example of printing. "The press were the first to draw the aesthetic consequences from the materiality of ideas, with a new art, whose concept had nothing to do with the traditional techniques; with the art of message composition, of layout (for example a newspaper), where the artist solves an aesthetic problem in the arrangement of communication fragments that they do not produce themselves. The clear division of editing and typesetting, the aesthetic irrelevance of that which is typeset, makes the materiality clear, and the discovery of other forms of communication confirms that this is not a coincidence, but rather a general peculiarity of the materiality of the communication of ideas; all in all this is nothing other than the artist's classic choice between assembling and producing: conductor and composer, painter and paint producer, architect and builder."[17] In the meantime, the distinction between *assembling and producing* has also been taken on board in professional software development, for example between the *software-architect*, who makes conceptual decisions, and the pure *coder*, who writes the programs. A variety of new job profiles arise in this way. We also find divisions of labour in artistic praxis, when artists work together with technicians and programmers for the realisation of their concepts. Otherwise the communication of ideas by means of formal codes is still quite complicated. Program codes are not only an extremely precise means of communication for computer scientists and programmers, but at the same time are already *instructions* for the machine.

The material properties of codes in use are different to those in production—in other words programing. In use, the material properties shift from the linearity of the code and its two-dimensional visual presentation to the materiality of the computer and its interfaces. Let us look at the example of so-called *warm sound* as a characterisation of sonic quality and *tonal beauty*. For classical musical instruments this is independent of the geometric dimensions of the instrument, conversely it is strongly dependent on the inner tensions of the material and its temperature. The warm sound of a pipe is therefore essentially determined by the wood used and not by its size. In the field of computer music, such knowledge of materials is transferred to a great extent to the interface. Material properties such as, for example, the frequency

[16] Timo Meisel, "Design und Medienwandel: Vom Medium Computer zur Theorie des Informationsdesign" (master's thesis, Universität Lüneburg, 2004).
[17] Ibid.

ranges in which a speaker breaks out into so-called partial oscillations, i.e. the membrane no longer moves piston-like in just one direction, are discussed there. But the materiality of the computer also interacts with the codes and creates the materiality of the sound. Two further examples on this: “Dust settles in the windway of the pipes, especially of small pipes, thus impairing the quality of their speech; to a large extent dust is the cause of ciphers and other mechanical failures. Therefore, the church should be cleaned only when the windows have been opened. The organ gallery must never be swept; rather it should be damp-mopped, possibly with the aid of moist sand or sawdust. Once every eight years, a conscientious organ technician should thoroughly clean all pipes and vacuum-clean the chests, floors, walks, and structural parts of the case. Humidity and dryness alike can damage leather and wooden parts of an organ; wind leakage in the forms of ciphers and runs, valve failure, and similar defects are commonly caused by such conditions. Moisture will also result in the rusting of metal parts. Churches with a high humidity content must often be thoroughly aired (on dry, not too cold days), especially after the floors have been washed or damp-mopped. Excessive dryness is often caused by the heating system. [...] Under direct exposure to sunrays wooden parts of the organ will warp or even crack. Heavy window drapes are the answer to this problem.” [18] The preceding recommendations for the correct treatment of material come from a reference work on church organs. Here we read of experiences about how the material that the instrument is made of directly effects its sonic qualities.

Another example from the world of program codes: “Crackling and popping: if your computer is not fast enough to compute the set number of voices at the set sampling rate in real time, the overload warning will normally appear (processor overload). In some circumstances, especially if the processor load limit (maximum processor usage in %) is set very high in the preferences, this can lead to dropouts in the audio signal, which can be heard as crackling and popping. Reduce the sampling rate, the number of voices or the complexity of the ensemble and observe the CPU load with the help of the meter in the toolbar or the system monitor.”[19] In both examples the material announces itself and instructions are given as to how the unwanted behaviour can be eliminated. Proper handling, maintenance and care of the material is necessary in order that artefacts do not malfunction. But both quotes also illustrate a further, very general characteristic of artefacts, namely that the properties of the material remain hidden in normal use and usually only come into question when the artefact malfunctions. Once again, we are reminded of the difference between *readiness-to-hand* and *presence-at-hand* that we were introduced to as a general design principle earlier. Heidegger’s bizarre concepts point to the difference between the practical interaction of the human with its environment and the theoretical treatment of things; that tipping over of an artefact in the moment of its refusal to function, the materiality of the thing once again resurfacing. The Heideggerian readiness-at-hand describes the yet unbroken relationship of the user to the thing, the condition of unquestioned trust. Through the fragility of things, through their refusal to function, they become in need of our attention and come back to light as present-at-hand. Only artefacts that malfunction leave the space of unreflective use and show their whole selves again. The ciphers and runs of the organ and the crackling and popping of the computer speakers are what makes an interest in the material necessary. The materiality of the instruments, which is made to disappear through successful design by the designer/developer, re-emerges in a malfunction. This breach is a phenomenon that primarily befalls the user; it does not exist in this acute form for the constructor, as they must always consider all perspectives simultaneously in the design and production process. The goal-oriented, smooth usability—readiness-at-hand—is only one facet of the artefact’s full picture. The good

[18] Hans Klotz, *The Organ Handbook* (Missouri: Concordia Publishing House, 1969), 123.
[19] From the troubleshooting appendix of the user manual for synthesiser and audio-processing system REAKTOR, version 3 (translation).

craftsman/artist/engineer can be distinguished by the fact that they can effortlessly change their perspective; they can shift the frame of references of their considerations and thus discover remote causes for the malfunction of the system; from initialisation errors in the code to a loose contact in a cable. As soon as the error has been found, the image reassembles and the consistency of the system is ensured again for a while. Through the occurrence and elimination of errors we learn more about the *nature* of our computer systems than is shown to the user in material-less ready-at-hand use.

But in what way do material and code interact in the second example in order to create the crackling and popping in the audio signal? Particularly as code on its own does not have its *own sound*. "What is digital media's own specific sound? Digital material itself does not have a sound; there is no physical material that is principally related with the digital code, so also no sound related to it. On the contrary, digital coding is an encoding of changeable agreements. Which physical form a digital symbol assumes is solely a concurrence of those involved in the coding and decoding. Digital code is not characterised by any particular materiality, but rather only through countability and indexation."[20] This analysis is absolutely accurate: code is distinguished by the changeable agreement and is not bound to a certain material. But in order to be readable and archivable, it must be imprinted in a material. So that the code can be processed, there must, furthermore, be a material configuration—a processor—that is capable of reading and carrying out the code. The agreement between code and material, which is necessary for their congress, is made by the system developers. Both—code and hardware—are already meant for each other in their design and only that which was always intended to, comes together in the execution. The computer's hardware is conceived in such a way that certain commands and classes of command sequences (programs) are realised through the material. This is the functionalism of informatics: the concrete code determines the functional conditions of the machine. As these functional conditions are realised in a material system and react to a specific input with a specific output, they update themselves in the material on the basis of the code itself. Likewise, the interfaces must be prepared in order to take in defined bit-streams and realise them in analogue material behaviour. The forms these streams can take are predefined. Just as codes have no sound of their own, conversely the material computer and interfaces do not have sound properties independent of code. First *informed material* as a combination of matter and algorithm exhibits sound properties.

If we characterise material primarily by means of its quality of use, then in the case of digital media, code is only one component of a compound of material, the other is the materiality of the computer and its connected sensors and actuators. We must, for example, change code parameters (for example reducing the sampling rate) in order to reduce the processor load and thus prevent crackling and popping etc. We change the code and get different physical behaviour. The code for controlling a motor that stimulates balls hanging on rubber strings to swing, can not be viewed detached from the physical properties of the strings, the weight of the balls and the characteristics of the motor (see *Connect* by Andreas Muxel). For this reason the discussion on material and material properties in connection with digital media is not spared, rather it spreads itself over code, interfaces and hardware. Working on code means having the material behaviour of the complete system in mind and controlling this behaviour using the code. In conclusion we can say: the fact that code has no sound of its own (colour of its own, movement of its own etc.), does not mean that we do not need to think about and take into account the material properties of our artefacts while coding. The material properties of digital objects are fused with the code. Just as important as the new properties that emerge is

[20] Rolf Großmann, "Spiegelbild, Spiegel, leerer Spiegel: Zur Mediensituation der Clicks & Cuts," in *Soundcultures*: *Über elektronische und digitale Musik*, ed. Marcus S. Kleiner et al. (Berlin: Suhrkamp Verlag, 2003), 59.

the *unique sense* newly formed from the combination of code and material. There is more information in each material than we are ever explicitly aware of. We must also assume we do not know all the relevant information in our daily dealings with things and therefore always expect even long trusted things to malfunction at inconvenient moments. Every manual activity thus requires a permanent analysis of the limits of predictability. But the unique sense of the new composite of code and material cannot be determined merely thorough contemplating the digital objects, rather only through practical work and empathetic observation.

This undertaking is difficult because two cultures meet in the algorithmic artefacts: the rationalist tradition, which appears to have reached a kind of (temporary) plateau with program codes, and the aesthetic tradition with its very own history of development rooted in both artistic praxis and philosophical reflection. Producing sophisticated digital artefacts means taking both traditions into consideration and constantly performing translation work in both directions.