# IMPLICATIONS OF UNFOLDING

Juilian Rohrhuber

In spring 2002, the scattered fragments of a meteorite entered the earth's atmosphere and plunged to the ground close to the 19th century Neuschwanstein Castle. The impact did not cause any damage, but was noticed by local observers and its trace recorded by astronomical camera networks.[1] After a reconstruction of the trajectory from the photographic data and some systematic search in the field, two pieces were retrieved on German territory. Some time later, following a corrected model of the expected shape, a third and largest fragment was found by a German physicist across the nearby Austrian border. Since these meteorites (of the enstatite chondrite type) are well recorded and of relatively high value, their material presence immediately caused a conflict between potentially rightful owners, such as the mayor of the small town of Reutte (by proxy), and the finder. The court case turned out to be intractable – the most heterogeneous categories of law had to be taken into consideration; after all, the laws that govern the interaction between heavenly bodies do not regulate the accumulation of wealth.

Seen from a distance, all that had happened was a minor extension to the planetary material. However, according to extant law, the situation could not be easily decided. Is a meteorite like snow (which 'falls' under the responsibility and property of the landlord), is it like apples from neighbouring premises, or like flotsam (which is regulated by specific laws)? Is it to be considered a natural monument? A report by the Bavarian state lawyer Kristine Faust discussed these issues circumspectly; she clarified that a meteorite is not material fallen from neighbouring premises, and that the ground it has fallen on has not produced it either.[2] Only something that is lost, can be found, and as the state had not acquired the meteorite in the moment of impact, it was not lost property either: despite the fact that gravity may be enough to juridically bind a thing to its premises, the meteorite was still light enough to be easily removed without the application of "disproportional effort". Yet with the first fragment, Faust came to the conclusion that the case was analogous to the discovery of hidden treasure, a solution that led to simply cutting the stone in two halves of equal weight, one for the state, one for the discoverers, who divided their half and sold the fragments. In the second case, the claim by the Austrian town was delivered a rejection, culminating in the statement "there is no earthly right to heavenly goods", and ownership was granted to the finder.[3]

1 The fish-eye camera propeller of the European Fireball Network scans a complete night sky every night. The photographic observation was published in Spurný et al. 2003, pp. 151-153.

2 Faust 2003, pp. 28-31

3 It remained unclear whether the then rightful owners had to pay income tax for their new possession.

Affiliations between things and living beings take the most diverse forms; this theme could hardly be more quotidian – everyday life is occupied with houses, tools, vehicles, but also with the material reality of weather and gravity, and trying to separate environment from inhabitant may at any moment turn out to become a knotty issue. Also, nothing stops us from extending the realm of things to languages, signs and symbols, which can forcefully turn out to condition an umwelt, just as they can assume the place of implements. So when it comes to the origin of actions and intentions, this context dependency makes it necessary to consider interactivity as fundamental to any investigation. There remains a certain dichotomy here though: while objects, structures, rules, or laws depend on a particular timelessness, movements, processes, interventions are almost exclusively temporal.

As the observation of scientific and artistic practices shows, it would be a mistake to locate the origin of attributes like intentionality or initiative in the human mind alone. Social, material, structural circumstances force decisions, just as they are subject to modification and investigation. It is possible to avoid a foundational choice between a social constructivist and a realist view by a different account of objectivity. This objectivity is the result of interaction. Such "situated knowledges", as Donna Haraway writes, "require that the object of knowledge be pictured as an actor and agent, not as a screen or a ground or a resource, never finally as slave to the master that closes off the dialectic in his unique agency and his authorship of 'objective' knowledge."[4] In this context, it is significant that many theories seem to have shifted agency toward objects instead of looking for it in processes. In Alfred Gell's anthropology of art, for instance, the main agent is the material art object.[5] For Donna Haraway, and also for Michel Callon and Bruno Latour it is the various hybrid, yet material coalescences that defy categorisation as either natural or social, animate or inanimate; they take the shape of collectives, 'agencements', which, despite their multiple forms, tend to crystallise in matter.[6] Hans-Jörg Rheinberger proposes an intermediate form between concept and object, the epistemic thing.[7]

Other than simply resisting common preconceptions, there are also good reasons for applying the notion of autonomy to material objects rather than to processes merely because the latter appear closer to 'being alive'. Firstly, objects imply specific actions and inherent necessities – they can be

4 Haraway 1988

5 Gell 1998

6 Haraway 1988; Latour 1993; referring to Donna Haraway, Callon writes, "These agencies, like Hobbes' Leviathan, are made up of human bodies but also of prostheses, tools, equipment, technical devices, algorithms, etc. The notion of a cyborg aptly describes these agencements." (Callon 2005)

7 Rheinberger 1997

described in terms of constraints and resistance, both physical and social. In a sense, things implicitly encode processes. A ball in a game is a classic case.[8] Secondly, an object occupies a place, may be attached to owners, can be passed on, and is thus able to transport action patterns between nodes in a network of relations. It may itself become a node, being equally the subject of, and subjected to, new formations. Not to mention that things can be traced, and sometimes collected. Nevertheless, according to agency theories, it would be wrong to treat objects (as well as subjects), as primary to their relationships with each other. Rather, it is the association between agent and patient that results in the resistance, the stubborn, 'objective' ignorance toward change that causes objects to exist, somewhat like the apparent stasis of an eddy or vortex in the flow of a stream. These linkages, which French sociology termed *operation chains*[9] are transactions of potential action.[10] In this capacity, they are agents with social leverage. Because they result from these chains, objects are inherently political.

The decision to put aside the essential polarity between the agency of persons and the agency of things allows us to treat the collective situation as existing logically prior to subject, action and object, and to render them conceptually indistinguishable. Deliberately creating a mode of observation with a blind spot for these distinctions causes new differences, subjects, and situations to appear. Areas of thought that tend to suffer when actions are merely considered as transmission from internal intention to external expression can be better explored by not presupposing objects to which intentions and actions of subjects can be moored. Thus, issues of intentionality in art and the intricate relation between discovery and fabrication that constitutes objectivity in the sciences cease to disturb investigation. In what way, for instance, does an artwork participate in the possibility of its own formation or condition its own becoming as it unfolds? How does a discovery turn out, after the fact, to constitute the very place it must have been part of already? In such issues, rather than a hindrance, paradox and undecidability turn out to be the driving force that opens previously inconceivable possibilities. "Experimental systems," Rheinberger writes, "[…] allow researchers to arrive at unprecedented, surprising results. In this sense, such systems are

8 For Michel Serres, the football is a good example for what he calls a 'quasi-object'. His figure ground reversal illustrates how agency and patienthood may swap places: according to this perspective, in a football game, it is not the players who control the object. Conversely, is it the ball that is the subject of circulation between stations, and the players follow after it. A quasi-object is only an object insofar as its movement binds a collective (Serres 1987; Roßler 2008).

9 The term *operation chain* (*chaîne opératoire*) was introduced by Leroi-Gourhan in 1964. It has its origin in archeology, where it had been developed out of the research on the action patterns in the fabrication of Stone Age tools, traceable from raw material, completed artifacts, and their chippings on production sites (Leroi-Gourhan 1964).

10 Schüttpelz 2008

'more real,' if you will, than ordinary reality. The reality of epistemic things is their resistance, their resilience, their capacity, as 'jokers' of practice, to force us to abandon preconceptions and anticipations."[11] In the following, I will discuss agencies that force the formation of a series of conjectures, of open hypotheses. Especially I will do so with respect to a kind of inconsistent hybrid between representation and unfolding, which may be called a 'model'. A detour into a specific praxis of computer programming, interactive programming, will provide evidence for a specific kind of interactivity typical for experimental systems; it can be traced back to an agency in the formal. Here, interactivity will turn out to imply a rather simple, temporal paradox; instead of looking for it in an immediate 'presence' of coupling, this observation will help to show how the unfolding of an investigation implies interactivity between its own history and future.

## Program, model, trap

Taking operation chains as causes underlying the formation of objects does not imply that these objects explicitly represent actions. Art objects, for instance, may be cunningly prepared in a way to cause a certain impression, and to inhibit others; they may function to impress the audience, or a patron; tools, or other objects usually imply certain actions, but are not self-explanatory; such things are a part, or a trace of an operation chain, but do not give access to the chain itself. Nonetheless, there are also many cases in which operativeness is combined with its description e.g. calenders, maps, plans, recipes, algorithms. A *program* is one example of a thing with such a double nature: it explicitly provides a plan together with a method to actualise it. The text of a program represents two processes at the same time: in the context of a given computer language, it activates a computational process, which may (or may not) produce results. Second, and this is what is supposed to qualify a 'good' program, it causes a human reader to understand, in one way or another, this process. A program can be regarded as 'operative writing'[12], as an assemblage of a possible process and its description.

As in any language though, one should not expect a transparent translation from formula to meaning, or a complete reflectiveness between process and description. The hybrid assemblage of operation-representation is necessarily incoherent. An example of a process that produces a description may illustrate this; since we have to describe this process of description-making, we can think of the 'simplest' case as a description which describes nothing but its own making. Such programs whose output coincides precisely with

11 Rheinberger 1997, p. 246

12 Krämer 1993

their own source code (*quines*) are often fairly hard to understand; in order to write a program that is a description of how to type out its own text, one usually has to construct a maze of means and ends, of 'use' and 'mention', of quoting and unquoting of quotes. This may be a clue that there are good reasons for the fact that the description of an operation chain resists superposition with the chain it produces. Here, the semantics of such a program is its code in a literal form.[13] Yet from a different perspective, it describes the process of *producing* this very code; in other words, it is because object- and meta-language interrelate that makes a quine difficult; in less reflective programs, where means and ends are more separate, this difficulty is not so obvious.

In a formal language, the semantics of a sentence, its meaning, is called a *model*. Also, a model of a whole language is all that can be 'expressed' in it. The model as such is a purely mathematical concept that usually refers to an abstract domain; yet at the same time, algorithms are, in a sense, the mechanical equivalent of a part of mathematical praxis. So it is justified to ask what a given program-text really means from a formal point of view – what is its model? With respect to our train of thought, three possibilities are obvious. Does it express (1) its *result*, which is the effect and endpoint of its execution? Does it express (2) the *process* that leads to this effect? This is not obvious at all. In fact, looking at the details of specification, this will always remain a slightly ambiguous issue;[14] what seems like a description of a result (a domain) may take on a more operative aspect (a process) in another situation; in other words, operation chain and its effect can never entirely be disentangled. And (3), we may have to take the program literally and see in it an inscription of the programmer's thought rather than a direct description of either process or result.[15] Usually, these three levels are arranged in an

13 Note that this may be any kind of representation, which need not be 'text' formatted in ASCII code. A quine in a visual programming language, for instance, would have to compute its own visual code as an image, without re-using parts of this representation.

14 This difference can be formalised in computer languages, but most languages do not do this. Even with a formalised semantics, semantics remains a matter of decision (For a thorough discussion on algorithmic equivalence, see Blass et al. 2008). Formal systems like pi-calculus explicitly encapsulate semantics into the system, which is passed around between agents. Computer semiotics, on the other hand, emphasises the process of meaning production as interactive coupling between cultural and algorithmic processes (Andersen 2003; Nake 2003).

15 Alternatives (1) and (2) correspond to two different understandings of formal computer language semantics: operational semantics refers to the computational steps of its process, whereas for denotational semantics (this term is a bit misleading) the process does not matter, as it refers only to its eventual outcome. Note that this outcome may, in turn, be a process, such as an interactive application. In most discourses, the meaning of a program for the programmer (3) is not considered separately. Nevertheless, emphasising the human reader and the communication of ideas, the concept of literate programming brought forth by Knuth (1992) and Iverson (1979) requires code to be taken more literally. Andersen (2003, p. 190) even takes formal semantics to be "the rules that we employ ourselves to read a piece of program," and the compiler a "machine execut-

instrumental relation; code simply expresses its result, not unlike a pocket calculator. This is because when a computation is fast and leads to some sort of unchanging entity, such as a number or the data of an image, the process can be thought to implode in the blink of an eye.

The situation is a little bit different as soon as the program is to describe a process that unfolds over time. In the above case we can say that a reckoner's operational agency (and labour), calculating by hand, is replaced, and hidden, in the rules of the formal system. This is not so obvious when the program's result consists in processes, possible behaviours, actions. Here, it is easy to lose the distinction between the program as a process that leads to another process, and this latter process (which is a program too). The 'application' replaces the 'program'; possible interactions with the application are identified as the 'behaviour of the program', so that now the assemblage between plan and process is solidified in a thing, such as an interactive application, but also as an interactive installation, augmented environment, etc. In the first case it was the effectiveness of the algorithm that made it possible to neglect the ambiguity between process and product, in order to command a view on the relation between timeless formula and its immediate computational result. Here, it is the interactive computation that, by representing the behaviour of a possible agent, replaces the semantic ambiguity. On the one hand, the fact that now computation happens necessarily over time, makes it more obvious that a program is an automaton, which works exactly in so far as it has been *abandoned* by its programmer. The model is left behind; instead of being the originator's operational result (like a painting is the artists product), it embodies the originator's formal ghost. On the other hand, this autonomy also introduces the above double meaning between processes, where the 'making of' interactive behaviour hides in the runtime behaviour itself. The model becomes objective because it is, to a degree, independent, it is an operational proof of its own unfolding.

This is a reason why attention has been drawn to the notion of modelling as an ideological notion, both from epistemology and computer semiotics. While semiotics tends to suggest a constructivist position instead, where programmers create a reality,[16] and the model of a program is a metonymical and metaphorical structure, within a materialist epistemology, programming (or formalisation in general) is taken as an experimental process with its own structural constraints and mathematical domain: "A formal system is a mathematical machine, a system *for* mathematical production and is placed within the process of this production."[17] Objectivity is not a function of behav-

able Representation of this." To regard programs as narratives, or as discursive media has become more widespread today.

16 Noble et al. 2002

17 Badiou 2007, p. 43

ioural resemblance of an artificial object with the natural object of enquiry, but it is to be found in the stubborn openness of formal systems. Precisely in so far as the model has been left behind, it is an agent, and its behaviour subject to discovery.

There is a well-known cultural technique, which differs from numerical calculation yet represent a type of reckoning that may help to clarify this issue, or at least may show a way into further investigation. In his 1996 article *Vogel's Net*, Alfred Gell gives an analysis of cultural praxis that bears many aspects of his concept of agency of artefacts which he published two years later. For Gell, animal traps are a peculiar kind of thing capable of more than it seems at first glance. Essentially, a trap is a mechanical implement of the hunter's ability to catch or to kill. Its reactivity is a model of the hunter's awareness, its mechanism is a model of cognitive competence. "It is, in fact", he writes, "an automaton or robot, whose design epitomizes the design of its maker. It is equipped with a rudimentary sensory transducer (the cord, sensitive to the animal's touch). This afferent nervous system brings information to the automaton's central processor (the trigger mechanism, a switch, the basis of all information-processing devices) which activates the efferent system [...]. This is not just a model of a person, like any doll, but a 'working' model of a person." Similar to Latour's example of the 'sleeping policeman', an object takes the place of a human in the enchainment of causes and effects. Yet, Gell notes that, at the same time, the trap is not only a model of a hunter. A trap is an altered environment – not so much as the hunter perceives it, but rather as a portrait of the animal's perceptual Umwelt. In order to catch, it is a model of an observer being caught. Gell notes that "[...] if we look at traps, we are able to see that each is not only a model of its creator, a subsidiary self in the form of an automaton, but each is also a model of its victim. This model may actually reflect the outward form of the victim [... or] the trap may, more subtly and abstractly, represent parameters of the animal's natural behaviour, which are subverted in order to entrap it. Traps are lethal parodies of the animal's umwelt."[18]

As a superposition of two models of complementary observers, I think it is plausible that the agency/patienthood of a trap resembles that of an interactive program. Its mechanism is a concrete abstraction, a passe-partout of its parameters. Like a trap, a program is "a model as well as an implement."[19] Program and trap both encapsulate a hidden, objectified plan that unfolds into a scene only at the appropriate circumstances. They are situated models of a silent, absent observer which is present in the observation of an observer. The conditionality and sequentiality of an algorithm come to a halt

18 Gell 1999, p. 200

19 Gell 1999, p. 200

when it snaps shut. But what is actually caught by a program? And who is the hunter really? Is there a moment of closure at all? The analogy could be a trap itself.

Apart from notable exceptions, the process of interaction, as it obtains between a running program and its environment, seems to have little of the sudden abduction by a mechanical implement. And when it is not meant to ensnare a potential person, but rather to give access to a new situation and unforeseen observations, it cannot be simply a portrait of either cognitive hunter or cognitive prey. Rather it appears to shift between an instrumental aspect, where computation simulates physical processes or enables communication, and a more oblique situation, in which a given causality is disturbed, where it is not clear what actions find continuation and which percepts are consequences of the local logic. One aspect is easily forgotten though, when looking at real-time interaction in art, or also in scientific simulations. Because, unlike in the early days of computing, human computer interaction today involves mostly the relation between algorithmic processes and users, the activity of constructing a program in the first place is taken as a preliminary means for creating interactivity. More precisely, if we follow the chains of causation in the loop between the various participants, we find that the algorithmic process is like a parallel world, only accessible through experimentation within the premises of this specific set-up. The mechanism's constructive preconditions stay hidden and become apparent only in the agency of its behaviour. Much of the critical effort within media art has been aiming toward bringing these conditions into discussion, making them accessible and contextualising them in the politics of things. This is a broad field, since the computational chain potentially pervades the situation just as much as cultural meaning passes through the networks of calculations. In order to reason about the conditions of interactivity, it is necessary to expose not only the model (be it process or result, or further chains of semiosis), but also the model formation within the interactive situation.

## Finding out

Together with the concept of interactivity comes the notion of real-time. The fascination of self-regulation in the early cybernetic discourses is implicitly connected with a coming to life of a dynamic continuum of becoming. Also, over most of the 20th century, digital systems were somehow always too slow – the resistance to interaction took place in an interval between input and output. A continuous labour to integrate computational systems into the environment, for instance in the form of scientific simulations in the workflow of a laboratory or in the form of interactive music instruments, led to the reduction of the delay in the loop to the point that it became sufficiently short to give the impression of a neutral presence of time. Yet, simultaneously, measurement and display became fundamentals, which continuous or discrete interactive processes were to operate upon. More precisely, we can state that real-time interactivity is predicated upon the idea of the *parameter*. Pivoting on approximations of real numbers, real-time computation consists of a network of connected streams, whose immediacy is mediated through a parametrised mask of measured movement, sensor information; often, a graphical interface for such applications gives access to interaction points by means of images of sliders and wheels. More generally, a parameter-space is the implicit frame of reference in real-time interaction.

Interactive programming has been taking the complementary approach: instead of writing an interactive program that exposes continuous parameters at runtime, it exposes the activity of parametrisation itself, and more generally, the construction of programs at runtime. A starting point may be a very small formula – for instance a sound algorithm, which generates a process that, by converting it into an alternating current and playing it over a speaker, can be listened to. Instead of now thinking about what parameters need to be exposed to external change, and building an interactive application that can be used later, the formula that describes the process is rewritten directly. Changes of the program's time-map figure as the medium of interaction. Therefore, in such a situation, it is not so much the parameter space that is subject to experimentation, but the program text itself. More precisely, as we shall see, it is the different semantic levels of a program, which become thematic again.

In experimental mathematics, methods of interactive programming have been used to investigate the relation between a program text (a formal expression) and its output, which may be a set of numbers, or other formal expressions; within application design, such methods allow iteratively improve computer applications. Usually, each version of the program text simply expresses one such relation – one description corresponds to one result. In the conversational approaches of the 1960s for instance, an incomplete program would ask the programmer questions until the result was found.[20] Now, as we saw before, this outcome may not have to be static. It also may be a process that exists only insofar as it unfolds and changes over time. In algorithmic sound synthesis, for instance, the program text describes something irreducibly situated in time; of course, one may record a sound wave and later play it back, jump around in it, or play it backward. Nevertheless, this does not touch the *relation* between an algorithm and its unfolding. Only after the fact, is everything data.

20 Matthews 1968; this procedure survives in today's terminal application, where program and programmer interact by turn-taking in the form of a dialogue.

Taking a closer look at the structure of interactive programming, it becomes apparent that experimenting with inherently temporal results requires the relation between description and model to change significantly. Generally speaking, this is because the program as a description cannot consistently represent the changes to this description themselves. It turns out that should a program be rewritable at runtime, we are confronted with a paradoxical situation. If we start the whole computational process from the beginning each time something is changed, then the formula really can count as a valid plan of its unfolding. But each new onset means that the change itself is no longer situated in the moment it actually happens relative to the ongoing process. The world ends and is recreated. Another procedure is to divide the program into concurrent parts, each of which can be changed individually, but which may interact over time. Then, when a part is changed, it is the local unfolding of this new part that affects the rest of the system. This sounds like a good solution, because then, changes to the code happen in the context of a continuous behaviour of the system. But now we are confronted with the fact that the description does not reflect the behaviour as a whole anymore, since different parts of the system must be understood relative to their different points of departure. Moreover, looking at the program in its entirety, an assemblage of parts, it is not always clear whether they are a description of how to put together other parts or whether they describe the behaviour of such a part.[21] As soon as one tries to integrate the rules into the interaction, their divergent interpretations (expression, process, or result) come into play. Of course it is possible, as a next step, to formalise the structural transition between different descriptions and represent those as the program. We then still have to decide which part of the text belongs to what part of the structure. Even more, the structure of these changes itself is again subject to the same problem if it is meant to be part of the interactive situation. So while some kind of segmentation is necessary, it is not generally decidable what belongs together and what is separate.

This symptom, which arises within interactive programming, allows us to consider the implications of unfolding on a more general level. As we have seen, the instrumental relationship between description and process, when it is supposed to allow an interaction in real-time, comes with a specific exclusion of the process of constructing the same system. If an *immediate* coupling

21 The moment of substitution of a part by a different part dissipates the general ambiguity of semantics of a whole program within its own parts. See e.g. Abelson/Sussman 1996, ch. 1.1.5: "Despite the simplicity of the substitution idea, it turns out to be surprisingly complicated to give a rigorous mathematical definition of the substitution process. The problem arises from the possibility of confusion between the names used for the formal parameters of a procedure and the (possibly identical) names used in the expressions to which the procedure may be applied. Indeed, there is a long history of erroneous definitions of substitution in the literature of logic and programming semantics."

of a system is desired, the laws that regulate this coupling cannot themselves be subject to interaction. In order to include these laws, not only has the idea of real-time to be relativised, but even more, the meaning of the system, the concept of a model becomes a matter in question. As long as we know exactly in advance what the meaning of a program is supposed to be (as long as we have a good specification, e.g. of its behaviour), however difficult, there is a possibility to line up the chain between description, computation and result, and construct an appropriate formalism. However, trying to find a new model requires a re-ordering of the whole situation, because interactivity with rules (and not with their parameters) entails a multiple split in the temporal domain; where time ceases to be a 'domain' that could be called 'real-time'. The ambiguity between things and their operative formation reappears here in form of conflicting levels of meaning, which can only partly be disentangled by situated decisions.[22] Interactivity – maybe as opposed to interaction – turns out to be an open temporal antagonism, a differend[23] on the verge of multiple temporal levels and multiple possible models. In a sense, this makes interactive programming thinkable as a miniature version of an experimental process on the border between formal and empirical methods, iterating between the different mechanisms of explanation, conjecture and failure. It exemplifies that when no cogent specification is given, and the subject of investigation is ambiguous, we cannot line up formalisation with interactivity in such a way that the former is only the necessary sacrifice to the technical, whereas real-time behaviour is authentic becoming. A situation in which it should be possible to *find out something*, the delimitation between thought and act has to be made amenable to reassembly. In other words, since it is not obvious to what degree the outcome is a construction of a new structure or a discovery within the current one, an investigation concerns the relation between free decision and strict deduction, or, from a different perspective, of the rational and the social.[24]

22 Inspired by Lyotard's article 'Time Today' (Lyotard 1991), in our paper 'Algorithms Today. Some Notes on Just-In-Time Programming', we have discussed this multiplicity of history in the context of a concrete system for interactive sound programming (Rohrhuber et al. 2005).

23 To give an example from empirical science: If we have an idea what parameters of some physical process may be relevant for a law, for some invariance, we can measure them (if we are lucky), and then test our equations against this data. Should either the experiment or the theoretical framework inspire us to see some other possible parameter, the measurement has to be done again. This is how the interaction between material culture of science and its concept formation is usually explained. (See e.g. Pickering 1995; Rheinberger 1997). In other words, prediction depends on the past, yet at the same time, the significance of past facts depends on their future effects. This causes interactivity to be necessarily situated in incommensurable orders, or incompatible law systems. In Lyotard's terminology, it can thus be considered a differend.

24 Longino 2001; already the early discussions on interactive programming show consequences that this has for the place automata should have in a process of reasoning. In the introduction to the proceedings of the conference Interactive Systems for

The temporal aspect of this problem can be found in the relation between rules and their unfolding. Formally, it can be described as the relation between a structure and a model; more precisely, between some kind of formal deduction laws together with basic assumptions, the axioms, on the one side, and on the other, some mathematical domain, e.g. sets. Evidence suggests that formalisation is not only the necessary precondition for an interactive, experimental, empirical investigation; rather it is already part of this investigation. The construction of a logic requires mathematical assumptions. Or, for instance, within algorithmic composition, programming is not just a technique for building synthesisers, but is part of the compositional process. Similarly, in scientific operations, the schematism is not a precondition for empirical confirmation, but both are part of a new form. We have seen that this hybrid, this mutual implication between formal apparatus and empirical praxis is not simply a fusion. Rather, the structure of an investigation must be regarded as a paradoxical interaction between what is possible and what becomes possible by doing the possible. This "dialectic of formalization"[25] unfolds in a mutual determination of what is given and what is found out: "every creation of thought is in reality a creation of a new formalization and at the same time this new formalization establishes a relation or takes part in an interaction with the particularity of what we are trying to express."[26]

The process of *finding out something* operates in-between a discovery of something that previously existed, forming the conditions of research, and, at the same time, the construction of a new situation that did not exist before, but reconditions what can be constructed.[27] The model is a linkage between a new possibility and a situated context in which it becomes unavoidable.

Experimental Applied Mathematics (Klerer/Reinfelds 1968), Klerer quotes Burton Fried, who had written one year earlier that the "utopian notion of a computer, which accepts the statements of a problem and automatically finds a way of solving it is clearly chimerical, save for those 'problems' whose structure has been thoroughly understood and for which methods of solutions are well known" (Karplus 1967, p. 169). Again, Klerer confesses that his own motivation, in contrast to the majority of the contemporary academic community, is "based on just such a utopian basis." He emphasises that, while the term interactive is difficult to define, "[…] we would expect more than in the old process of inputting a well-formulated set of directions with the machine performing in its capacity as an idiot servant." (Klerer/Reinfelds 1968, p. 9). I think that quite conversely, Fried's comment alludes to the basic incompleteness of a majority of formal systems, showing that exactly because we may extend a given system by a term that is not derivable from within it, interactive programming is interesting. One cannot decide in advance what part of the system will turn out to be involved in such a change; this suggests interactive programming (of whatever kind) as an interesting conceptual alternative to interaction with a program that has its interaction point already defined in advance. So problem solving is indeed "chimerical", since it involves ambiguous agencies, which explains the need to include the programming activity itself into the program.

25 Badiou 2007, pp. 90-92

26 Badiou 2007, pp. 90-91

27 In this, the dialectics of formalisation are equivalent with Badiou's later concept of a truth procedure. Following Cohen's mathematical technique of forcing, Badiou is able to

## Traps revisited

Computation follows a strict protocol laid down in its description, yet at the same time gives rise to completely unexpected things. This is why an algorithm may, under some circumstances, occupy an intermediate position between a law as discovery (when we find it, we suppose that it has always existed) and the law as a constructed artefact (which is an intervention into what exists).[28] It becomes a hybrid between the ghost of the programmer, stood in for by the automaton and the autonomy of a new situation.

Let's return to the question in how far interactivity of this kind is structured in analogy to a trap. What is caught in an algorithm? Who is it an agent for? I think the interesting central thought that Gell started out with – namely that in some way, the trap is a model of both hunter and hunted – is useful for clarifying the situation of a 'dialectics of formalisation' that implies some paradoxes of interactivity. In his investigation of agency, Gell is able to show that certain situations and artefacts cause the observer to enter into a process of reasoning.[29] Their questionable mode of fabrication, the unclear origin, maybe we can say their artificial and alien character, causes them to force an attribution of agency. "Is this spot here on purpose or did it just happen unintentionally?" or "Is this strange sound we just heard part of the composition, or is it a mistake of the performer?" Agency in this sense is essentially an open question provoked by a disturbance of conventional inference, a question which can only be answered by hypothetical reasoning, or, as Gell puts it, abduction of possible originations. This search for the inner logic of a situation enmeshes the participants in possible alternative worlds of causal, and thus, temporal connections.

In a peculiar dialectics, agency is what an observer infers of a phenomenon's origin, and simultaneously it is the power to induce this reasoning. Like the trap, the artwork is a disturbance of the causal milieu in which it is situated,[30] a disturbance that opens an explanatory gap, a cognitive dissonance. We can say it is a different model of cause and effect within a given reference frame, a model that forces one to hypothesise about possible explanations (this abduction is the derivation of a law from a model). In this way, certain artefacts are able to induce interactivity – interactivity as a process

show ontologically under what conditions a new formalisation is possible in a given situation (Badiou 2007b). For an investigation of a relation between set theory and agency theory in this light, see Rohrhuber 2008.

28 The issue unfolds in a contradiction between place, finder (an investigation), and ownership: does the found object belong to the place or the finder? Does the place belong as much to the landlord as the object belongs to the finder? Does observing an event and investigating its traces set a rupture that contradicts the continuity of territory?

29 Gell 1998

30 Gell 1998, p. 20

of reasoning, of situated thought. That human beings may make such inferences is without question. But it is more interesting to ask what could be the conditions under which a new formalisation, a new causation may appear at all. As we have seen, in such a rupture, social and natural causes become mutually exchangeable, just as much as the difference between construction and discovery of reality have to be negotiated anew. So if an artefact, or more generally, a situation as a whole may be the cause of such a shift, how does it have to be structured? What is a model for finding models?

As a conceptual starting point, traps have turned out to be interesting as an epistemic model; they allowed us to consider something like an objectified anticipation together with an objectified ignorance.[31] At first, this was thought as two parts: an open conjecture and an automaton, a 'materialised theory' on the one side; and the unfolding scene of captivation on the other, where the enclosure must have already been entered before it snaps shut. It is in evidence that these, in turn, imply two concepts of time; in the first, the absentation of the hunter causes the prey's possible presence. In the second, the prey is already caught before it realises this fact. Like in a weir, for a fish there is no point where the difference can be found between inside and outside. In other words, in order to discover something, and not invent it, this entity must have some autonomy; however artificial the situation, it must show itself. Yet it must show itself in the situation that is given already. However if it is not certain what is to be caught, this separation becomes unstable, and as a consequence, the trap begins to resemble an experimental system. Here, the model breaks; hunter and prey become indistinguishable. Anticipation becomes a conjecture about a possible new situation in which we are entangled already.

Thus, the trap remains a model for the possibility of finding out (something). It suggests that it is the dynamics of laws that abstract from immediacy and allow experiment: the inherent temporal logic here is its formal indifference to time; just as abstraction allows statements not to differentiate between certain things, the abstract also shows indifference to the moment at which events occur. As we have seen, this is what made a program imply both possible actions and their formalisation, both operation chain and plan. But instead of giving rise to interactivity in the sense of a presence of unification, it has turned out that abstraction – if it is, despite all contradiction, included in the situation – leads to a resilient and antagonistic assemblage.

31 For Blumenberg, quite in accordance with the view Gell proposes, traps are intimately related to the peculiar temporality and agency in concept formation. He considers preemption in its dialectics between absence and presence: "The trap acts in place of the hunter in the moment of his absence, but in the prey's presence. These conditions are revealed to be the reverse in the trap's production. It is the reified expectation. Insofar, the trap is the first triumph of the concept [Begriff]." (Blumenberg 2007, p. 14, my translation).

Suspending direct access, a formalised situation may itself impel the process of a different form. The abstraction from the trap allows us to maintain that hypothetical situations do exist: they are not just a product of an observer, opposed to a non-hypothetical world. There are cases where abduction is to be found within the situation – finding out such cases is itself a matter of formal experiment (it must then be possible to find something that was not even hidden).

## References


Abelson, Harold/Sussman, Gerald Jay (1996): »Structure and Interpretation of Computer Programs«, Second Edition, Cambridge, London: MIT Press. Online available: <http://mitpress.mit.edu/sicp/full-text/book/book.html> (last access 6. July 2008).

Andersen, Peter Bøgh (2003): »Semiotic Models of Algorithmic Signs«. In: Karl-Heinz Rödiger (Ed.), Algorithmik – Kunst – Semiotik, Heidelberg: Synchron Wissenschaftsverlag der Autoren, 165-193.

Badiou, Alain (2006): Briefings on Existence: A Short Treatise on Transitory Ontology, Albany: State University of New York Press.

Badiou, Alain (2007): The Concept of Model: An Introduction to the Materialist Epistemology of Mathematics, Melbourne: re.press.

Badiou, Alain (2007b): Being and Event, London: Continuum.

Blass, Andreas/Dershowitz, Nachum/Gurevich, Yuri (2008): »When are two algorithms the same?«. Online available: <http://lambda-the-ultimate.org/node/2729> (last access: 6. June 2008).

Blumenberg, Hans (2007): Theorie der Unbegrifflichkeit, Frankfurt a.M.: Suhrkamp.

Callon, Michel (2005): »Why Virtualism Paves the Way to Political Impotence. A Reply to Daniel Miller's Critique of Laws of the Markets«. Economic Sociology 6(2), 3-20.

Faust, Kristine (2003): Wem gehört Neuschwanstein?, München: Aviso, Bayerisches Staatsministerium für Wissenschaft, Forschung und Kunst.

Gell, Alfred (1998): Art and Agency: An Anthropological Theory, New York: Oxford University Press.

Gell, Alfred (1999): »Vogel's Net«. In: Alfred Gell, Eric Hirsch (Eds.): The Art of Anthropology: Essays and Diagrams, London: Berg Publishers, 187-214.

Gell, Alfred (1999b): »The technology of enchantment and the enchantment of technology«. In: Alfred Gell, Eric Hirsch (Eds.): The Art of Anthropology: Essays and Diagrams, London: Berg Publishers, 159-186.

Haraway, Donna (1988): »Situated knowledges: The science question in feminism and the privilege of partial perspective«. Feminist Studies 14, 575-599.

Iverson, Kenneth. E. (1979): »Notation as a tool of thought«. In: Communications of the ACM, 23(8), 444-465.

Karplus, Walter J. (1967): On-Line Computing, New York: McGraw-Hill Education.

Kay, Alan C./Goldberg, Adele (1977): »Personal dynamic media«. Computer 10(3), 31-44.

Klerer, Melvin/Reinfelds, Juris (1968): Interactive Systems For Experimental Applied Mathematics, New York: Academic Press.

Kneer, Georg/Schroer, Markus/Schüttpelz, Erhard (Eds.) (2008): Bruno Latours Kollektive. Kontroversen zur Entgrenzung des Sozialen, Frankfurt a.M.: Suhrkamp.

Knuth, Donald. E. (1992): Literate Programming, Stanford: Center for the Study of Language and Information.

Krämer, Sybille (1993): »Operative Schriften als Geistestechnik. Zur Vorgeschichte der Informatik«. In: Peter Schefe (Ed.): Informatik und Philosophie, Mannheim: BI-Wissenschaftsverlag, 69-84.

Latour, Bruno (1993): The Pasteurization of France, Cambridge: Harvard University Press.

Leroi-Gourhan, André (1964): Le geste et la parole (2 Volumes), Paris: Albin Michel.

Longino, Helen E. (2001): The Fate of Knowledge, Princeton: Princeton University Press.

Lyotard, Jean-François (1991): Time Today, Stanford: Stanford University Press.

Pickering, Andrew (1995): The Mangle of Practice: Time, Agency, and Science, Chicago: University of Chicago Press.

Rafaeli, Sheizaf (1988): »Interactivity: From new media to communication«. In: R. P. Hawkins/J. M. Wiemann/S. Pingree (Eds.): Sage Annual Review of Communication Research: Advancing Communication Science: Merging Mass and Interpersonal Processes 16, 110-134.

Rheinberger, Hans-Jörg (1997): »Experimental Complexity in Biology: Some Epistemological and Historical Remarks«. Philosophy of Science 64, Supplement, 245-254.

Rohrhuber, Julian (2008): »Mengenlehre«. In: Ilka Becker/Michael Cuntz/Astrid Kusser (Eds.): Unmenge: Wie verteilt sich Handlungsmacht?, München: Fink Verlag, 141-156.

Rohrhuber, Julian/de Campo, Alberto/Wieser, Renate (2005): »Algorithms today – Notes on Language Design for Just in Time Programming«. In: Proceedings of the 2005 International Computer Music Conference, San Francisco: International Computer Music Association, 455-458.

Roßler, Gustav (2008): »Kleine Galerie neuer Dingbegriffe: Hybriden, Quasi-Objekte, Grenzobjekte, epistemische Dinge«. In: Georg Kneer/Markus Schroer/Erhard Schüttpelz (Eds.), Bruno Latours Kollektive: Kontroversen zur Entgrenzung des Sozialen, Frankfurt a.M.: Suhrkamp, 76-107.

Schüttpelz, Erhard (2008): »Der Punkt des Archimedes. Einige Schwierigkeiten des Denkens in Operationsketten«. In: Georg Kneer/Markus Schroer/Erhard Schüttpelz (Eds.), Bruno Latours Kollektive: Kontroversen zur Entgrenzung des Sozialen, Frankfurt a.M.: Suhrkamp, 234-258.

Serres, Michel (1987): Der Parasit, Frankfurt a.M.: Suhrkamp.

Spurný, Pavel/Oberst, Jürgen/Heinlein, Dieter (2003): »Photographic observations of Neuschwanstein, a second meteorite from the orbit of the pribram chondrite«. Nature 423, 151-153.