A process-oriented approach to the science of human-computer interaction

Mathieu Magnaudet, Stéphane Conversy, Stéphane Chatty

To cite this version:

Mathieu Magnaudet, Stéphane Conversy, Stéphane Chatty. A process-oriented approach to the science of human-computer interaction. [Research Report] RR-ENAC-2019-01, ENAC. 2019. hal-02104183

HAL Id: hal-02104183
https://hal-enac.archives-ouvertes.fr/hal-02104183

Submitted on 19 Apr 2019

Equipe Informatique Interactive

Mathieu Magnaudet, Stéphane Conversy, Stéphane Chatty

RAPPORT DE RECHERCHE

RR-ENAC-2019-01 19/04/2019

A process-oriented approach to the science of human-computer interaction

2019-01

A PROCESS-ORIENTED APPROACH TO THE SCIENCE OF HUMAN-COMPUTER INTERACTION

MATHIEU MAGNAUDET, STÉPHANE CONVERSY, AND STÉPHANE CHATTY

Abstract. Since the birth of the field, HCI has defined itself both as a theory of the relations between humans and numerical systems and as a practical activity that aims at building new interactive systems. However, HCI has not yet succeeded in discovering a unified theoretical framework nor in building a strong link between both activities. Based on an analysis from various fields, we show that most of the difficulties come from the computational paradigm that is still used as a foundation of most of the theories in HCI. This brings us to proposing a new philosophical view on the science of HCI, based on a process ontology. We show how it accounts for several phenomena related to HCI and unifies them. This approach lends itself to new ways of thinking and programming interaction at different scales, which may help HCI scientists in their modelling and design activities.

1. Introduction

Since its early days in the 1960s, the discipline of Human-Computer Interaction (HCI) has been defined with a twofold objective: being a scientific discipline able to describe, predict and explain the relationships between humans and computers, and being an engineering discipline aimed at designing better interactive computing systems. This clearly shows in the ACM definition [34]:

> *Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.*

Although it is difficult to find comparable definitions of other scientific fields, this project is by many aspects analogue to the articulation between physics and engineering physics or between chemistry and chemical engineering i.e., the derivation of engineering methods and processes from a scientific corpus. However, the specificity of HCI is that there is not yet a firm scientific body of knowledge upon which to build an engineering field. There is not such a thing as a unified, uncontroversial, theory of human-computer interaction. This diagnosis has already been made, notably by Yvonne Rogers [66]. Her review of the literature from the last 30 years reveals an heterogeneous and overabundant set of theoretical frameworks. Moreover, according to her study, it appears that the professionals involved in the design of interactive computing systems do not base their activity upon any of these theoretical framework but rather on a set of recipes and good practices.

We as a community could accept this fact and just consider that the object of HCI is so complex that we cannot hope to build a strong unified theory, in which case we could satisfy ourselves with inventing new interaction techniques and building partial and local theories. But beyond intellectual discomfort, the lack of firm theoretical background and the impossibility to derive effective design methods from a theory have practical consequences. The first one has been clearly captured by Greenberg and Thimbebly with the expression "weak science" [31], by which they mean the difficulty to manage a growing body of knowledge, the difficulty to replicate experiments, and that of having clear evaluation criteria for assessing the value of a contribution to the progress of knowledge. Thus, a recent study shows the apparent inability of the HCI community to focus on specific, structuring, research topics [43]. Not surprisingly, it seems that this theoretical weakness has some consequences on the academic recognition received by the community [73].

A second consequence is the difficulty to build an educational program for students in HCI design. To start with, what is the need to teach them the various existing theoretical frameworks if they are of no use in their future practice? Then, if there is no agreement about which theory to teach, how can we do more than just collecting good practices?

A third problematic consequence is the difficulty to reuse HCI innovation in industrial processes, especially in the development of safety critical systems. As long as it confines to an empirical discipline, HCI cannot fully satisfy the current strong demand on proof of safety or formal verification of interactive systems.

Solutions have been proposed to tackle these various issues, ranging from unification attempts [22] to the derivation of design methods from a theoretical framework [9, 20]. In a more pessimistic move, some researchers even suggest to give up any general theory and to create an intermediate level of knowledge, somewhere between theory and practice [37]. While we share the diagnosis of most of these papers, our approach here is noticeably different. We propose a philosophical analysis that aims at improving our understanding of the very object of the science of human-computer interaction and, consequently, to outline what could be the principles for a theory of this object.

In the first part of this paper we recall that the situation has not always been so unfavourable and we come back to the initial intentions behind the science of human-computer interaction and its underlying paradigm, the computational theory of the mind. Then we present some of the criticisms addressed to this paradigm and we show that they all point at its inability to account for major phenomena in HCI as well as in cognitive science and even in computer science. In the third part, from a thorough analysis of the object of the science of HCI, and the observation of an ontological shift in contemporary science, we suggest a new theoretical direction based on the concept of process. Finally we explore some possible consequences of this shift for the science of HCI.

2. Cognition and computation

There are at least two ways to tell the history of HCI. The first way focuses on technology and browses the successive inventions in the field and the new interaction techniques that they enable. Brad Myers' brief history of HCI, for example, takes this approach [56]. The second way focuses on the various theoretical trends from the 1960s to nowadays [66]. This is the one which interests us here. Our thesis is that the theoretical crisis in HCI is strongly related to a crisis in its initial structuring paradigm i.e. the computational approach to cognition.

2.1. **The origins.** As suggested by its name, human-computer interaction borrows most of its theoretical content from two fields: the social sciences, specifically psychology, and the sciences of the artificial [74]. But the two apparently independent bodies of knowledge combined in HCI have a common root. This root is in the cybernetics movement that emerged just after World War II [88], and, one step upstream, the seminal 1936 paper by Alan Turing [82]. This history is well known [30, 33, 25] but it is worth recalling a few aspects of one of the major contributions of Turing's paper, the cross-comparison between the brain and the computer.

While originally intended as a contribution to the foundation of arithmetics, Turing's paper had a wider audience than just mathematicians. Indeed, his formal work on computable numbers is the first convincing formulation of an equivalence relation between the operation of the human mind and that of a logical machine. In all honesty, Turing did not tell very much about the human mind and it was not his goal. In particular, he never said that the mind is embodied in a machine whose functional architecture is isomorphic to a Turing machine. But Turing's thesis establishes a weak equivalence relation by setting that for each task that a human can perform with a pen and a paper, his logical automaton can produce an equivalent result. In other words, for a given input, a human calculator and a Turing machine will produce the same result.

It is only few years later, in 1943, that a stronger thesis has been formulated [53]. In their paper, McCulloch and Pitts proposed a computational model of how the brain operates. By taking neurons as a set of very simple calculators connected in a network, they tried to show that this machinery has a computational power equivalent to that of a Turing machine . Thus, they provided the missing link that allows to reason about the human mind as a kind of computing machine. Not only the Turing machine can do whatever the human mind does, but the brain itself can be modelled as a logical automaton. This hypothesis, explored by the cybernetics movement during the famous Macy Conferences, provided a basis for the foundation of a new materialist science of the mind. In a critical paper, John Searle summarised its basic principle: "the mind is to the brain as the program is to the hardware" [69].

Of course, the cybernetics movement is richer than this simple statement. It provided many seminal ideas about control loops, feedback and auto-regulation mechanisms, to cite a few. However its main heritage in contemporary cognitive science consists in this computational metaphor that allowed the constitution of a truly structuring paradigm.

2.2. **The computational paradigm.** We can find various expressions of the above equivalence relation in all fields constitutive of cognitive science: cognitive linguistics [17], cognitive psychology [63], cognitive neuroscience [18], philosophy of mind [62], and computer science [57, 58].

One reason for the popularity of this thesis is probably that it allows the scientific study of an object, the mind, that was formerly reserved to metaphysics or religious thinking. Philosophers of cognitive science use the expression "naturalising the mind" [24] to account for this process of appropriation of the mind by natural sciences. But, more than a simple theory, this approach founds a paradigm i.e. a framework that structures a research field in the Kuhnean sense [45]. Indeed, it provides the delineation of a phenomenal field, a set of concepts to categorise it, the formulation of a set of problems that must be solved, and a specification of the possible forms that answers to these problems can take. In summary, it brings all that is needed by scientists and engineers to structure their daily activities.

It is important to note that the computational paradigm cannot be reduced to the computational metaphor, as it also conveys a specific way of thinking about computers and computational processes. The key concept, also derived from Turing' works, is that of algorithm. Within the paradigm, the scientific problems are expressed in algorithmic-related terms. This clearly appears, for example, in the work of David Marr and Tomasio Poggio on the neurophysiology of vision, with their proposal for a three- or four- level analysis of cognitive processes [52]. According to them, at the upper level scientists start their investigation by formulating the *computational* problem that the cognitive process is solving. This can be done, for example, by finding an arithmetic expression that specifies the mapping between a set of inputs and a set of outputs. One level below, scientists have to hypothesise an algorithm allowing to compute this function. Finally, they have to find the mechanism, or the functional architecture, that implements this algorithm. In a similar approach, Zenon Pylyshyn proposed an algorithmic-oriented level decomposition to guide the research in cognitive psychology [63].

The computational paradigm rests on the concepts of problem solving, function, input-output mapping, programs, algorithms, functional architecture to cite a few. Brain or psychological phenomena must be thought out as transformation processes from a set of inputs to a set of outputs. And some legitimate questions are: what are the various functions realised by a cognitive system? What is the algorithm that would allow computing this function? What is the functional architecture that would realise this algorithm?

The computational paradigm also implicitly defines norms and criteria about what is an acceptable explanation of a cognitive phenomena. In particular, one of the basic assumptions of this paradigm is that if we can build a computational model of a cognitive process then we know at least one machine able to realise it i.e. a computer. As Jerry Fodor [28, p. 13-14] explained, the Turing-calculability of a cognitive model is a safeguard against magical explanations of mental phenomena. This warrants that we stay at all times within the boundaries of a natural science of the mind.

2.3. **The computational paradigm in HCI.** The role of the computational paradigm in the constitution of the field of HCI is easily understandable: if the humain mind and the computer are thought of as two machines of the same kind, able to execute the same functions, then we can consider the possibility to substitute or to augment one by the other. This is the path taken, for example, by Licklider and Engelbart in two classical papers known for their influence in the field [48, 26].

Note, however, that if we talk about human-computer interaction, it is because the substitution is not complete. The reasons invoked for this incompleteness are various. Some argued that our knowledge about human cognition is insufficient or that computers are not powerful enough. Others, like Licklider, had a more skeptical stance about the very project of a complete substitution such as expressed in the field of artificial intelligence. It seems that Licklider never believed in the strong equivalence between mind and computing machines. But everyone seems to agree that many tasks can be made easier through a kind of interaction between humans and computers, and this opened the door for a completely new research program. On the one hand, there is the issue of process distribution between humans and computers, following the seminal work of Fitts and his MABA-MABA list [27]: who is better at doing what? On the other hand, there is the issue of interaction techniques or interaction modalities: what is the best way for a human to interact with a computer? And this paves the way toward the invention and evaluation of new interaction modalities.

The influence of the computational paradigm over the field of HCI appears more clearly with the concept of cognitive architecture. Cognitive architectures, such as SOAR [46] or ACT-R [1, 2], are executable models describing the alleged functional architecture of the human cognitive system. They are strongly related to the current progress in cognitive psychology, which refines the decomposition of the cognitive system in functional modules. They are also used within the field of HCI: Newell, Card, and Moran, for example, developed a human processor model [15] that serves as a theoretical basis for GOMS and KLM.

Historically, the computational paradigm strongly shaped the way of conceptualising Human-Computer Interaction. Indeed, if humans and computers are two kinds of cognitive systems, and if a cognitive system is a symbolic information processor, then human-computer interaction is just an information-processing related activity. The complete human-computer system transmits, stores, handles information. Thus, human-computer interaction is primarily thought as an informational interaction or, as proposed by Brey [13], as an epistemic interaction, i.e. a knowledge exchange.

Classic task analysis is another consequence of this way of thinking the human mind as a kind of computational system. A task can be split up in goals and subgoals and its realisation can be distributed over computers and human processors. This is the very principle of task analysis and task allocation. As Sheridan [72] puts it: *"we are particularly concerned with allocation between humans and intelligent machines - computers - where the capability of the modern machine now comes closer to that of the human than in the past".* Thus, task distribution between humans and computers rests upon the idea of, at least, a weak equivalence relation between the one and the other.

Task allocation does not mean a complete separation between human and machine, each one doing its job separately. Conversely, according to Sheridan, there is a complete range of possible cooperation modes. However, the interaction model that stems from this analysis is the question-answer schemata such as illustrated by Turing's imitation game[81]: each system gives an input to the other; the computer may furnish a piece of information, a list of possible options for example, while the human may enter a list of commands, such as the selection of one of the proposed options.

At this point, one may ask whether this model can account for any kind of interaction between a human and a computer. What about, for example, the continuous move of a finger over a sensitive surface, or the handling of a joystick or a steering wheel to maintain a system parameter within its safe boundaries? At first sight, there is no obvious way to describe such situations within the question-answer schema. Of course, the computational paradigm produced many rich and interesting results that we do not intend to contest. But one can question its ability to provide an adequate descriptive and explanatory framework for every phenomena surrounding human-machine interaction. As we will now see, the issues are deep and numerous.

3. The computational paradigm against its criticisms

The computational paradigm received many criticisms since its appearance in the 1950s. Some of them led to a more or less sophisticated refinement of existing theories. Others adopted a more radical stance and called into question the basic principles that we presented above. We analyse here some of the criticisms from three theoretical fields: HCI, the philosophy of cognitive science, and theoretical computer science. These criticisms are well known by the researchers of each discipline, but they are not always known outside their respective fields. By putting them together, we intend to show their convergence toward the idea that the computational paradigm cannot account for a wide range of phenomena surrounding human-machine interaction, thus calling for new concepts.

3.1. **Situated cognition.** Suchman's work [77] appears as the first departure from the computational paradigm. She explicitly points the cognitivist approach, an instantiation of the computational paradigm within the field of cognitive science. More specifically she criticises the idea that human action should be understood as the sequential execution of a plan previously defined and decomposable in elementary sub-actions. She puts forth the fact that action is always situated, and that the process of decision making is localised and strongly depends on specific circumstances. Decision making rests on the dynamic interactions with the people and the objects constitutive of the immediate context of the action.

Briefly stated, this line of criticism provoked three kinds of attitudes within the community. The first, directly inspired by Suchman's work, consists in a renunciation of the computational paradigm and in the rejection of cognition as an object of study. The theoretical framework becomes that of the social sciences, especially ethnology, and the primary aim is to develop new design methods to improve the usability of human-machine systems. Many works about participatory design seems to stem from this intellectual shift [51] as well as works on embodied cognition [23].

A second reaction to Suchman's criticisms consists in reinterpreting them within the computational paradigm like Vera and Simon did [86]. In a similar vein, the works of Hutchins [39] and Kirsh [41, 42] offer some extensions to the classical paradigm in order to account for these criticisms. But they retain the idea that human action is "cognitive" and that it involves steps of computation and information processing even if some of them are delegated to the environment. Thus, their works appear as fully representative of the so-called "extended mind" view of cognition [19].

Finally, the third kind of reaction to Suchman's work amounts to minimizing its impact on the field of human-machine interaction. The continuing works about cognitive architectures [50] appear to us as a clear illustration of this line.

The influence of Suchman's work is undeniable and the various reactions to it partly explain the diversity of the theoretical approaches in HCI. However it did not succeed in providing the basis for a new paradigm, perhaps because it was too narrow in scope and did not provide any insight on some issues studied in HCI. One can cite, for example, the necessity to develop formalisms to assess the safety of a critical system or the need for programming tools dedicated to the development of interactive systems. But, at least, the literature on situated action reveals that human-computer interaction cannot be fully understood if we reduce it to an informational interaction between two abstract computational systems. Something more is required to account for the complex network of interactions constitutive of human use of computing devices, the pending question being whether we have to complete the computational approach with some exogenous theoretical content or to proceed to its complete replacement.

3.2. **The dynamical approach to cognition.** Within the field of cognitive science and philosophy of mind, the criticisms against the computational approach are numerous and varied. Some of them are close to Suchman's, such as the embodied approach to cognition [85, 79]. In a different attempt, some researchers tried to show that the capabilities of the human mind exceed those of a Turing machine, and consequently that we cannot reduce the one to the other (cf. [49] and more recently [60]). Others insisted on the difficulty for this paradigm to account for some alleged properties of mental states such as intentionality [69] or phenomenal consciousness [12]. But in the beginning of the 1990s a new type of criticism emerged, accompanied with the proposal of a new paradigm: the dynamical approach to cognition [61].

One of the reasons motivating this new trend is the thesis, coming from the situated and embodied approaches to cognition, that cognition cannot be understood from the unique study of the intra-cranial processes. Cognition should be better viewed as the result of the complex interactions between brain, the dynamics of the body, and environmental variations [10]. In addition, a strong emphasis is put on the temporal dimension of the interactions between these various systems. According to its supporters, the concept of time from computability theory and reduced to the number of steps necessary to solve a problem, is not suited to account for the continuous changes characterising cognition [83].

"Dynamicism" thus proposes to appeal to another, more adequate formalism: the mathematical theory of dynamic systems. This theory rests upon a qualitative approach to differential equations for which there is no analytic solution [35], and is focused on the evolving behaviour of the system along the time. It allows to characterise some of its properties such as attractors (fixed point, limit cycle, strange attractor, etc.) or bifurcations. The adoption of this point of view over cognitive phenomena has important consequences. They are no more seen as functions producing an output from an input through an algorithm. Dynamicists are more interested in the asymptotic behaviour of the cognitive system, its properties of stability, its convergence toward a specific state, or at the opposite its oscillatory behaviour. A dynamical approach to decision making for example, shows the oscillatory nature of the mental states before the convergence toward a specific choice. As put by Townsend and Busemeyer, the dynamical approach succeeds where the classical algorithmic approach fails in its ability to describe "the variability and the temporal evolution of preferences" [80].

In defence of their theory, the proponents of this approach often argue that it introduces a continuity between the usual vocabulary of the natural sciences and those of the science of cognitive phenomena [84]. This is actually a desirable property; however this also points toward a weakness of this approach. Indeed, it seems difficult to build a structuring paradigm out of the simple request to adopt the formalism of dynamical systems. The dynamical approach is quite silent, for example, about which cognitive variables should be modelled. More generally we hardly find the expression of a concrete research program with a set of questions and the specification of a methodology to study them (however see [16] for an attempt). As a consequence, from an HCI point of view, it provides no immediate resource to build a design framework.

Despite this limitation, two points deserve our attention. First, the dynamical approach sheds light on the inability of the computational paradigm to account for some properties of cognitive phenomena, especially their temporal structure. Second, it reinforces the idea, already set by the situated, embodied approach that there is a need for a unified theoretical framework to understand and model the interactions between phenomena of different nature.

3.3. **Process algebra and reactive systems.** Up to now we focused our analysis on the role of the computational paradigm in the field of cognitive science. But it is interesting to note that criticisms also appeared in the domain of theoretical computer science at the end of the 70s, notably with the works of Hoare [36] and Milner [55].

Theirs was not a criticism about the validity of the classic computation theory but about its ability to account for some properties of communicating systems. Indeed, computers quickly moved from the initial computing machines with a single central processing unit toward more complex systems equipped with multiple processors and various communicating sub-systems acting in parallel. With those systems, a new range of problems appeared related to the interaction between processes. We can cite for example the issue of the distribution of processes over multiple processors, their synchronisation, or deadlocks and concurrent access to resources. It appeared that the classic computation theory did not have the conceptual tools to reason about these problems, thus the need for new formalisms such as process algebra [6].

Beside these works, also appeared the concept of reactive systems [11] and the related issue of the engineering of systems that react in real-time to multiple asynchronous sources of events. More recently, the works about cyber-physical systems also stress the importance of introducing a different concept of time in computing processes [47], a demand consonant with that of the dynamical approach to cognition. In both cases, this reveals the need to build new abstractions and new languages to cope with some problems ignored by the computation theory.

Note that neither process algebra nor reactive systems theory intend to exceed the expressive power of the Turing machine such as is sometimes suggested by the proponents of the concept of hypercomputation [21]. The idea is rather to address other issues than the characterisation of the set of computable functions, the classification of formal automata or the study of algorithmic complexity. This is a matter of filling in the explanatory gap of the theory in a similar way as those exposed in the previous section.

At the opposite, Wegner produced an important work to characterise the concept of interactive system. In asserting their superiority over Turing machines [87], he stayed within the classic computational paradigm and did not propose a new one that would better fit the range of concerns related to them.

4. The object of the science of HCI

The previous analysis calls for a new structuring theoretical ground to step aside from the computational paradigm. However, we cannot propose a new framework without a clear characterisation of the phenomena at stake. So what exactly is the object of HCI?

4.1. An endless diversity.

The diversity in HCI is not only in the successive conceptual frameworks that have been elaborated in the field, but also in the studied phenomena and the scientific questions about them. When looking for the objects studied in HCI, we find a combination of design spaces exploration triggered by the appearing of a new technology, software architecture studies, basic researches on human cognition, action and cooperation, investigations on engineering methods, etc.

As a consequence, there are several ways to describe HCI. At some point it was a branch of computer science that possibly borrows some content from other fields with one goal: designing interactive computing systems. This is probably still the common conception in the other scientific fields, that see HCI as an "applied" computer science. One can also see it as a composition of computer and cognitive sciences as originally conceived by the computational paradigm. This view has been pregnant in various industries. Finally, one can see HCI as the science of the interaction between human and computers built from the knowledge produced by other fields, thus focused on the relation rather than on the relata.

This last approach has been adopted by many researchers of the field. However, this characterisation does not go without difficulties. Indeed, it puts a special focus on the human-computer channel, which easily brings back to the computational paradigm as a common ground between humans and computers. This focus also tends to exclude from theoretical frameworks any surrounding phenomena such as the other physical systems that may interact with the computer and/or the human user. But more importantly, the properties of the human-computer channel are continuously evolving with the introduction of new technologies: HCI started with data forms, dramatically changed with graphical interfaces, then discovered multi-modality, augmented reality, and finally tangibility that brings it to rediscovering the interaction between human and physical objects.

However beneficial it is to the field, this closed link with technological innovation regularly causes the import of new phenomena and new conceptual frameworks that were previously considered as foreign: physiology of perception, signal processing, the physics of force feedback, holography, and so on. Every interaction phenomenon, be they electromagnetism, gravitational fields, chemical reactions, exchanges of information between algorithms or human relationships can possibly play a significant role in HCI. Thus, it seems to be a real challenge to maintain a unified conceptual framework for the field.

4.2. **The science of interactions, for designers.** To cope with this difficulty, recent works in HCI have pointed out the need to focus on interaction and to provide a definition that fits the need of HCI studies [38, 76]. We fully agree that interaction is the real object of HCI, but we still have to define its theoretical goal and to structure its method of investigation. There is a need of a conceptual framework that accounts for each interaction phenomenon that may occur in a human-machine system, in a unified and systematic way, still avoiding to be the simple addition of the science of each of these phenomena.

It is worth recalling that HCI studies actually involve three stakeholders: an artefact, a user, and a designer. However, they do not have the same status. The common ground to all actors of the community is the interactive artefact. This is what guides the development of the field and its boundaries. Human cognition, for example, is of interest to HCI only as far as it is involved in the use of computer-based artefacts. In the same way, skin chemistry will presumably become interesting for HCI as soon as we enable humans to interact with skin [90].

Thus, not only do we need a conceptual framework for describing interaction, but it is also required that this framework be usable in the design process of a computer-based artefact. This is where the designer jumps in: the role of the designer in HCI is to shape interaction, to "program" it. Of course, as a science, HCI expects to have some descriptive theories about interaction, one of its model being the famous Fitts Law [32]. But, considering its practical dimension, a theory in HCI is important only as far as it can be used by a designer. HCI aims at providing computer-based objects designers with every model and theory that allowing to plan the interactions between these objects and their environment. This covers the programming i.e. the definition of the future behaviour of the system during the design process, and the modelling to analyse the behaviour of already existing systems. In our view, the role of the designer in the use and evaluation of theories what makes HCI different from most scientific disciplines.

5. Thinking interaction

As we have characterised it above, HCI inherits traits from natural and human sciences, and from design, computer science and engineering. From the two former it inherits the description of interactions, from the two latter the concern for supporting designers. In order to develop a proper theoretical contribution, we thus propose that HCI moves toward a general theory of interaction. HCI must be able to go beyond the individual sciences and to account for the phenomena in an homogeneous way so as to reduce the gap between the way we describe a computer-based interactive system and the way we build it.

5.1. **The need for an ontology.** The quest for homogeneity has been a recurrent concern in the history of science especially in the 17th century. One way to address this issue is to set up an ontology, i.e. to propose a categorisation of the basic constituents of the world. At the most abstract level, when considering ontologies we are faced with two main choices. The first is a substantialist ontology, and the distinction between the substance and its attributes such as found in the philosophy of Plato or Descartes. Natural sciences strongly rely on this distinction between objects and attributes, and on this uniformisation of nature. It is the basic principle behind our contemporary view that the world can be fully described as a complex configuration of atoms, themselves thought as very small objects. Moreover, this ontology fits well with our ordinary experience of the objects as persisting and countable entities, localised in space and time, to which we attribute properties. Substantialism also inspired Alan Kay when he designed the object-oriented approach in computer science [40].

However, other kinds of ontologies exist that put in the first place the concept of process [71, 65]. Indeed, our experience of the world is not simply made of objects and their attributes. Many sentences of the ordinary language do not fit in substantialist ontologies: "it rains", "the room temperature is decreasing", "the fire is propagating quickly", etc. These sentences do not have a well-defined entity, thing or person, as their subject; they denote processes i.e., these aspects of human experience mainly characterised by changes. As we will try to show, processes provide an adequate theoretical background for a proper theory of interaction.

5.2. **The case for a process-based ontology.** Focusing on processes leads to stop focusing on the precise nature of the elements that are interacting. The point of interest becomes the structure of change and the structure of its propagation from a process to another one, that is the structure of the interaction process itself. For instance one focuses on a flow as such (its temporal and spatial evolution), without considering what is flowing: water or lava. This is also a way to give precedence to the cause-effect relationship, or to the continuous or discrete nature of a process, over the specific nature (the attributes) of the phenomena at stake.

In the technical vocabulary of philosophy, the passing from a substantialist ontology to a process-based ontology is a conceptual shift. Such a shift has already occurred in contemporary physics. Quantum field theory, one of the most supported theories in fundamental physics, does not find any easy interpretation within a substantialist ontology. And indeed, many of the paradoxes of quantum physics appear solvable if one gives up the substance-attribute schema in favour of an interactionist view of properties [44].

A similar shift occurred in biological sciences. For instance, neuronal synchronisation [75] aims at understanding and modelling the interactions between processes distributed over several parts of the cortex. The structure of the processes thus prevails over the nature of the involved elements. Similarly, works about the modelling of ecosystems take as their main object of study the networks of interactions between living organisms as can be seen in the presentation of the review *Ecological Modeling* [1].

While the dynamical approach to cognition described above does not explicitly refer to it, we assume that it calls for such a process-based ontology when it describes the phenomena of change and the interaction between these changes. In the same way, process algebra in computer science considers processes as first class citizens, exclusively characterised by their interactions with other processes. There again, the main theoretical objective is to formalise the relationships between processes: parallelism, synchronisation, concurrence.

We claim that HCI, because it takes interaction as its primary object of study, should follow this shift.

6. Principles of a process-based analysis of interactive systems

In the following we will try to show how a process-based ontology can be used as a basis for a new structuring paradigm for the field of HCI. By this we mean that, like the computational paradigm, this new paradigm can be used to circumscribe a phenomenal field and to structure its scientific investigation.

At the most general level, we define a *process* as "any expression of change": the activation of a specific user's brain area, a finger moving on a sensitive surface, the change of the visual appearance of a press button, and so on. The concept applies to every part of an interactive system and can be used at any abstraction level. The user of an interactive system can be considered as a single process compounded of (a possibly large number of) sub-processes, down to the chemical reactions in her cells and to physical particles. Likewise a computer can be considered as a process encompassing a complex entanglement of simpler ones, down to the flow of electrons through the metal. The boundaries and the granularity of the processes differ according to the system under study: the design or modelling of a system supporting collaborative work will not focus on the same processes as would the design of a new tangible interactor. Selecting the scope and level of analysis is a designer's or modeller's choice.

Studying individual processes is the matter of various sciences: physics, chemistry, mechanics, etc. HCI as we mean it should not be interested in isolated processes, and should not intend to be the science of each processes constitutive of an interactive system. The phenomena of interest should be the causal relationships between processes: a gesture that triggers an update of the graphics, the hand-eye coupling in target acquisition tasks, interface adaptation (e.g. keyboard backlight) to environmental change (e.g. luminosity). Following the process theory of causality elaborated by W. Salmon [67] for physical explanation, we consider the causal influence between processes as a propagation of activation. In the following we call it a *coupling.* A coupling can itself be described as a process, that

---

[1] *We aim to understand these basic ecosystem functions using mathematical and conceptual modelling, systems analysis, thermodynamics, computer simulations, and ecological theory. This leads to a preference for process-based models embedded in theory with explicit causative agents as opposed to strictly statistical or correlative descriptions.*

changes the activation state of a process according to that of another process. *Reciprocal interaction* can be interpreted as a chain of couplings that creates at least one loop.

A process may have some features that determine the possible couplings with other processes. A time-varying process can be described according to basic categories such as oscillation, continuity, sequence of phases, etc. Couplings with other processes rest on these properties. For example, the blinking behaviour of a visual alarm may activate a human thought process, while the simple display of the alarm without blinking may be ignored. This property of a process can be understood from its structure, that is the complex arrangement of sub-processes and couplings that composed it.

According to this paradigm, the analysis of an interactive system turns into the analysis of the features and structure of the processes at stake, and the understanding of the link between these structures and the coupling between processes. Regarding the practical side of the HCI field, that is the design of interactive systems, the main concern becomes the specification of the processes that one wants to be activated or to evolve at any point of the interaction, the identification of the expected couplings, and the implementation of the control structures that will enable them in an artefact.

7. Revisiting the HCI body of knowledge

The conceptual shift that we suggest here does not imply giving up the huge amount of work that has been done in the HCI field. On the contrary, one of the most important evaluation criteria for a candidate paradigm is its ability to provide a framework for reusing the existing knowledge. We propose here to show how the HCI body of knowlegde can be revisited with a new, process-oriented, point of view. To this effect, in the following we present selected models from HCI and propose an interpretation within the new paradigm. To ensure that the diversity of HCI is reflected, we chose examples from nine out of the twelve sub-committees of the CHI2019 conference Programme Committee, leaving out only three that are dedicated to specific areas.

7.1. User Experience and Usability.

One of the key topics in usability is situation awareness. Alarms are of particular interest in the design of safety critical interactive systems. Some data is available on the efficiency of auditory, visual and haptic alarms; we show here how this fits in the proposed paradigm and can help to monitor the design process of alarms. Alarms are supposed to warn users of something that is happening in the background, so that users can take actions if necessary. Using the process-oriented paradigm, one can model this by considering all the systems involved, and describing their interactions. This includes a phenomenon process (e.g. the landing gear of an aircraft that improperly deploys), a process that monitors it with a probe (e.g. a cylinder position tracker), a process that displays an alarm (e.g. a blinking LED), the human perception processes (visual, in this case), and the process of executing a well-defined procedure to solve the issue. Designing alarm systems consists in providing ways to couple these five processes, that is four couplings.

From a hardware and software point of view, it is crucial to ensure that there exists a first coupling between the physical process and the probe. Then there are three major HCI problems with alarms: their reliability, their perception, and the trust that humans put in them. A false positive or a false negative can be considered as a failure of the conditions of the activation of the second coupling between the probe and the alarm display. A non-perceived alarm can be considered as a failure of the third coupling between the alarm display and the user. This depends both on the attention level of the user and on the design of the display and its coupling compatibility with human perceptual means: possible bad designs could be a single change of colour in peripheral vision [8], or a continuous (non-evolving) sound in a noisy environment. Finally, the training of the operators aims at building the fourth coupling between the alarms and well-defined operational rules[64].

Similarly, the level of autonomy of so-called “self-driving” cars can be described according to the distribution of processes between the automated systems and the driver [59]. In a level 0 car, the driver maintains multiple processes to steer and regulate speed, to monitor the direction and the traffic. In a level 1 car, some processes (speed) are maintained by automated systems. The driver is required to run monitoring processes and to maintain a physical coupling between her hands and her foot and the input devices to take over control (assuming that the existing automatic couplings will be disabled in case of emergency i.e. a process that deactivate the automatic steering or speed control process). In a level 2, steering is handled by automated systems. The driver is still running a process to monitor the traffic and the behavior of the car and a process to maintain a physical coupling. By comparison, in a train the steering process can be considered as the result of the coupling between the rails and the train wheels. Finally, in a level 3 car, it is the responsability of the car (a process of the car) to restore a coupling with the driver whenever the car self-assesses its own inability to perform safely, usually by an alarm.

The above descriptions are rough models of the situations. For instance, they do not represent the multiple levels of monitoring and questioning that can occur in human brains, the training processes, nor the subparts of physical systems and their possible behaviors. But one benefit of the process-oriented description is these models can be refined or enriched by adding new processes and subprocesses to reach the desired level of accuracy: this suggests that the approach might be generative.

7.2. **Learning, Education and Families.** Pilot training is a very interesting example of pedagogical challenges. In the case of flight Rio-Paris accident in 2009, the authorities recommended training sessions for the rarely-met events that led to the accident[14]. Various processes can be identified and studied. For instance, training is a means to couple an alarm process to the process of applying a maneuver or procedure. However this coupling is implemented through human memory, that degrades over time through a process than can be modeled. Trainers are therefore faced with a choice: either implementing a memory refresh process through rehearsal sessions (a new process that must be implemented), or using intensive initial training so as to turn the coupling into a skill, with a much slower degratation process. Another option could be to ask cockpit designers to find another implementation for the coupling, using a more explicit alarm or an alarm that triggers a natural skill.

The benefit of the process paradigm in this case is the homogeneity it introduces between phenomena of very different natures: alarms, skills, training. Reasoning in terms of processes allows analyses and dialogues among experts from different backgrounds.

7.3. **Interaction Beyond the Individual.** Air Traffic Control involves multiple human actors, co-located or remote, synchronously. To start with, controllers often work in pairs. At this level, their activity is made of information processing (analysing situations), but also of communication, interruption and monitoring (warning the teammate about a problem, choosing a visual or speech modality according to his level of activity). For describing this activity in order to support it, the concept of process can be directly transposed from computer science. But beyond mere description, processes can be used as the basis for architecting future air traffic control systems. For instance, introducing automated support can be analysed as the selection of processes that can be migrated from humans to the computer, and as the creation of new monitoring processes.

As for the dialogue with aircraft, it can be described at high level as a collection of couplings between ground and air systems, in addition to RADAR displays. The design of future aeronautical systems architectures, with one pilot only per aircraft, with the supervision of several drones by one operator, or formation flights across oceans, becomes an exercise of assigning processes to pilots, controllers and computers, and creating the appropriate couplings. For instance, when a controler gives an instruction to a pilot, the monitoring process she starts to check that the aircraft is complying will probably be the same with a purely automated aircraft: the delegation process is overall the same. Reciprocally, some processes currently implemented onboard aircraft will probably be implemented on the ground and the question will be where it is most efficient to implement them, between air traffic control centers and airline operational centers. Analysing the cost of creating couplings with other ongoing processes will help make the decision.

The benefit of the process paradigm here is that the analogy that is frequently seen by computer scientists between distributed systems and collaborative systems finds a natural expression, without having to reduce one to the other.

7.4. **Privacy, Security, and Visualization.** There are multiple ways to enforce security on mobile devices: from digicode, to touch gesture or fingerprint recognition. The IPhone “FaceID” allows users for unlocking their phone by recognizing their face. FaceID requires to couple the user and the camera of the phone. When this coupling is active, it triggers the unlocking of the phone, and triggers a timer that will deactivate the coupling when the phone is not used during a given period. This interaction assumes that the user is looking at the phone in front of it, which might be true most of the time. However, there exist situations where one wants to unlock the phone without being in front of it, for example as a jury member when turning on a timer when the defense of a dissertation starts. In this case, a more inconspicuous interaction is needed, e.g fingerprint recognition. Again, there should be no difference when describing the whole system, whatever the identification means.

7.5. **Accessibility and Aging.** Accessibility characterises the usability of a system for persons who have different abilities. This characteristic can be formulated in terms of processes. To start with, the abilities of the users and the various interaction styles can be described in terms of processes. For instance, an aging person will have degraded couplings that can be described with delays or with a lower probability of propagating activation (e.g. because of lower attention span). Usability can be described as the compatibility between processes, and evaluated through the probability and the time to reach a given state.

In addition, asking what kind of additional support can be provided to, e.g., elderly users can be reduced to asking what additional processes must be designed, for instance a monitoring process to detect incomplete interaction sequences and trigger alarms. With this regard, the connections between accessibility and safety-critical systems is easy to establish with process-oriented analysis: the same redundancy patterns are present and can be verified.

Finally, from the engineering point of view, accessibility often requires the ability to substitute one process by another, for instance s static display with an animation, a gesture by a sequence of button presses, etc. Being able to reducing these heterogenous mechanisms to a homogenous concept reduces the engineering complexity.

7.6. **Design.** The activity of designing interactions necessitates three processes: understand the needs, ideate/synthesize, and assess the design. User-centered design stipulates that users be considered as the central focus by designers. Designers should first understand the needs by analyzing users, then ideate and synthesize the design, and finally come back to users. Participatory design stipulates that users be part of the designers. This shortens the cycle, and better ensures that new features stick to users' needs. Presumptive design gets rid of the understanding aspect, replacing it by fast, disposable protototypes to make users react often and guide the design[29]. The differences between the three approaches can be described with increasing amount of coupling between users and designers, and increasing amount of couplings between the three processes understand, design, assess.

7.7. **Interaction techniques, Devices and Modalities.** Being able to describe interaction techniques unambiguously and completely is essential to the field: not only can it prevent implementation errors and facilitate teaching, it also allows to analyse and compare the properties of techniques. With this regard, the Keystroke Level Model can be interpreted as relying on a clever serialization of the processes that describe common interaction styles. Thimbleby's work on medical devices can then be interpreted as the use of other mathematical tools (e.g. graphs) to compute properties of interaction techniques from a formal description of processes [78]. In this context, the use of parallelism, state machines or data flows to describe interaction techniques can be paralleled to their use in the SysML language to describe processes in systems engineering: the same syntaxes are used in two different domains, the common ground being un underlying process-based description. Parallelism, state machine and data-flows are examples of organising collections of couplings in well known patterns or control structures.

One can also use processes and couplings to analyse how interaction modalities match the use context. For instance, Virtual Reality (VR) goggles isolate the user from the outside world, breaking some couplings between the user and the world In contrast, Augmented Reality (AR) goggles superimpose artificial imagery on top of the real world: the user is still able to perceive the surrounding world. In other words, the user's eyes are still coupled to real objects. AR also adds coupling between real objects, virtual ones, and the user: positions, or actions that can be triggered with gestures on real or virtual objects. AR is thus more about adding interactional capability instead of computational capability.

7.8. **Understanding People: Theory, Concepts, Methods.** Fitts' law has given birth to many research on how to beat it[7]. For example, McGuffin and Balakrishnan [13] (hereafter M&B) suggested a way of facilitating pointing with no permanent spatial cost by temporarily expanding the target during the end of the approach. They found that performance benefited from target expansion even when the target only began to expand as late as after 90% of the movement toward the target. However, M&B massed their static and expanding targets in separate blocks of trials, thus making expansion predictable for participants. Zhai&al (ZCGB) replicated their experiment with one new condition in which the target could unpredictably expand, shrink, or stay unchanged. ZCGB results show that target expansion occurring as late as in M&B's experiment enhances pointing performance in the absence of expectation.

A process-oriented account of these experiments can be done as follows. M&B condition is the simple addition to the canoncial 1D Fitts task of a process that monitors the distance between the cursor and the target, and a coupling that triggers a scaling by two of the target width when the distance reaches a given value. ZCGB unpredictable condition is the same process, with a random choice among 3 scaling values (2.0 - expand, 1.0 - static, 0.5 - shrink).

Theoretically, M&B's findings supports a long line of thinking in human motor control. An old theory postulates there is a two-phase theory in rapid reaching movement - ballistic and current control [89], with the first phase being open-loop and the second closed-loop. A more recent theory is that of [54], which postulates an optimized trade-off between the first and second phase. Since only the second phase involves feedback control, it is plausible that a target that expands just before the second phase of control is begun is as good as a target that has that larger size permanently. Given that human closed-loop reaction time takes about 100-200 ms [68], it wass surprising that one can still take full advantage of the expanded target at so late a stage (90%) in movement execution.

In other words, two processes govern the task: a vision process, and a hand-control process. The 'open-loop' theory stipulates that there is no coupling between the two processes during roughly half of the gesture, while a coupling between the eye and the hand is established to aim at the target at the end of the gesture. The 'close-loop' theory stipulates that the coupling is present all along the gesture. In both cases, the coupling adds a delay (100-200 ms) between the two processes.

7.9. **Engineering Interactive Systems and Technologies.** While it is not unusual to find a process vocabulary to describe physical or cognitive phenomena, we do not find any equivalence in the field of computer science, apart from very specific works cited above. However, our process-based analysis of human-machine interaction also suggests another view point over programming. Indeed, our claim is that the aim of the HCI designers is to build links between processes (cognitive processes, bodily movements, environmental changes, computer processes, etc.). In other words, they want to enable and to control the couplings between processes.

This view delineates a crucial mission for the field of HCI, that is the elaboration of the languages and tools to model and control causal links between processes in computer-based systems. At the programming level this notably consists in elaborating a process-oriented language equipped with control structures explicitly dedicated to the handling of causal links. When the designer specifies a relation such as “each time the user moves the cursor over the button, its appearance has to change”, the programmer should have at her disposal a simple control structure reflecting it, something allowing to express “when Process A then Process B”. Following our process-based ontology, this language should allow A and B to be any kind of processes: a sensor, a processor clock, an user input, a software process, etc. Moreover, this control structure should itself be construed as a process, that is as something that can be active or inactive. With all these features, such a control structure is the equivalent of a coupling as we defined it above. It can be complicated so as to account for the variety of causal relationships: fork, when Process A then Process B and Process C, conjunction, when Process A and Process B then Process C, probabilistic link, full determinism, etc.

Various kinds of control structures should be elaborated according to the specificities of the processes that one wants to connect through a coupling. If one wants to react to a continuous signal such as a change in luminosity or in temperature, one needs a control structure akin to a data-flow operator, that is a process that propagates change in a continuous way. To the contrary, if one wants to react to a pattern, with a specific ordered sequence of events (e.g. a finger touching a tactile surface, moving on it, then leaving it), one needs a more complex control structure such as a finite state machine.

We have shown in a recent work [4], that all these control structures can be derived from the basic control structure expressing the coupling between processes. As a consequence, it becomes possible to build full-fledge interaction-oriented programming languages. These approach and languages have been proved to be plainly usable for designing complex interactive systems [3, 5].

8. IMPLICATIONS

We have shown how the use of a process ontology to describe all parts of an interactive system makes them comparable, and fixes their apparent “impedance mismatch”. The alarm example involves hardware, software, human perception, learning and rule-based actions, all described in the same manner. This homogeneity facilitates the substitution of one process by another one, possibly of a different nature, provided that it has the same structure, thus allowing equivalent couplings with other processes or possibly enabling new ones. The “fly-by-wire” flight control system is an example of the replacement of mechanical processes by electronic ones. While most of the couplings are preserved, new ones become possible such as that with an automatic system to enable auto-pilot functionalities.

Moreover, by focusing on the couplings, the analysis may unveil those that have been lost during such a substitution and that could weaken the whole system. Thus, in the fly-by-wire example, force feedback may be artificially reproduced to maintain a physical coupling with the co-pilot, which may be important for shared situation awareness. This may play an important role in an iterative design process where process substitution (input modalities, visualisations, procedure, etc.) is a common practice.

Another important aspect of coupling is that of its "quality": its strength (the extent to which it influences the destination process) and robustness (the extent to which it holds despite other weaknening processes). For example, human vision is a coupling that can be impaired in case of smoke, or during the night. The motor/vision coupling can be strengthen using "beat-the-Fitts-law" techniques : in this case, speed and errors are dimensions of measurement of the quality of the coupling. This open the way for a quantitative evaluation of the couplings, and thus of the whole interactive system.

One could wonder how such an approach could help HCI designers in their activity. By removing the accidental complexity due to the impedance mismatch, it might be easier for them to understand the system as a whole, and eliminate design problems hidden by the mismatch. Since the approach can also be directly operationalised into code, it might be easier for them to implement interactive systems, and iterate in parallel models and their implementation.

9. Threats to validity

To assess the internal validity of our apporach, we gave some examples on how it can be used to model phenomena related to interactive systems. Even if the examples cover a significant span of HCI concerns (9 out of 13 subcommittees), a lot of work remains to be done.

First, the present analysis does not show that most of HCI concerns is actually covered. More work needs to be done to assess how the approach actually captures interesting aspects of other HCI concerns, and how this leads to interesting insights or to interesting conceptual tools for designers.

Second, the analysis might be not deep enough. Granted, we argued that the depth of the analysis may be varied according to the scale at which designers want to reason, but it remains to be proven with more examples that such method would give interesting outcomes.

Third, the approach is mainly descriptive as-of-today, even if some of it might be generative as it gives insight on how a particular process could be replaced by another one. Nothing prevents the use of current models and theories when they provide more powerful modelling (e.g. predictive), in parallel to the presented approach. However, work remains to be done to make the approach more powerful e.g. in a predictive and possibily quantitative manner.

Finally, to be effective, a more complete process ontology must be carefully elaborated. We have seen how the structure of a process may determine its causal influence. To understand these relationships, it is crucial to elaborate a complete set of concepts allowing to characterise, compare, or classify, the various kinds of processes. An inspiring work could be that of J. Seibt [70]. She proposes to provide the ontological foundation for any research program that wants to give the first place to the concepts of "process" and "interaction" through a conceptual framework to analyse the properties of a process. For example one may be interested by its property of auto-similarity, the fact that a (temporal) part of the process of raining is still a process of raining, by difference with a symphony whose parts are not themselves a symphony. More generally, we need a vocabulary, possibly formal, to describe the varieties of the spatial and temporal properties of processes.

10. Conclusion

Rejecting the computation paradigm, we proposed an approach to human-computer interaction by taking as the basic building blocks the concepts of process and coupling. We showed how such an approach can be exploited for the modelling and the design of interactive systems, and how it can be used as the theoretical background for the elaboration of an interaction-oriented programming language. Much work remains to be done to confirm and consolidate this theoretical proposal, which goes from formal work on the basic process ontology to its application to the various aspects of human-computer interaction. But we are confident that this theory could provide to the HCI community with a clear conceptual framework, with a unified set of concepts and a common approach to scientific issues. Such a framework might enable to structure the variety of works done in the field, and would provide a basis for discussion and collaboration.

References

[1] John R. Anderson. *The Architecture of Cognition.* Cambridge, MA: Harvard University Press, 1983.

[2] John R. Anderson. *Rules of the Mind.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1993.

[3] anonymous. What should adaptivity mean to interactive software programmers? In *Proceedings of the 2014 ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '14, pages 13–22, New York, NY, USA, 2014. ACM.

[4] anonymous. Djnn/smala: A conceptual framework and a language for interaction-oriented programming. *Proc. ACM Hum.-Comput. Interact.*, 2(EICS):12:1–12:27, jun 2018.

[5] anonymous. Vizir: A domain-specific graphical language for authoring and operating airport automations. In *UIST '18: Proceedings of the 31th Annual ACM Symposium on User Interface Software and Technology*, 2018.

[6] Farhad Arbab. Computing and interaction. In Dina Goldin, Scott A. Smolka, and Peter Wegner, editors, *Interactive Computation*, pages 9–23. Springer Berlin Heidelberg, 2006.

[7] Ravin Balakrishnan. "beating" fitts' law: Virtual enhancements for pointing facilitation. *Int. J. Hum.-Comput. Stud.*, 61(6):857–874, December 2004.

[8] Lyn Bartram, Colin Ware, and Tom Calvert. Moticons:: detection, distraction and task. *International Journal of Human-Computer Studies*, 58(5):515 – 545, 2003. Notification User Interfaces.

[9] Michel Beaudouin-Lafon. Designing interaction, not interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 15–22, New York, NY, USA, 2004. ACM.

[10] Randall D. Beer. The dynamics of brain-body-environment systems: A status report. In Paco Calvo and Toni Gomila, editors, *Handbook of Cognitive Science: An Embodied Approach.* Elsevier, 2008.

[11] Gérard Berry. The foundations of Esterel. In C. Stirling G. Plotkin and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner.* MIT Press, 2000.

[12] Ned Block. Troubles with functionalism. In C. W. Savage, editor, *Perception and Cognition: Issues in the Foundations of Psychology*. University of Minnesota Press, 1978.

[13] P. Brey. The epistemology and ontology of human-computer interaction. *Minds and Machines*, 15:383–398, 2005.

[14] Bureau d'Enquêtes et d'Analyse. Final Report On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris. Technical Report BEA f-cp090601, République Française, Ministère de l'Écologie, du Développement durable et de l'Énergie, 2012.

[15] Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction.* Hillsdale: Erlbaum, 1983.

[16] Anthony Chemero. *Radical Embodied Cognitive Science*. MIT Press, 2011.

[17] Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, Massachusetts, 1965.

[18] Patricia S. Churchland, Christopher Koch, and Terry J. Sejnowski. What is computational neuroscience? In Eric L. Schwartz, editor, *Computational Neuroscience*. MIT Press, 1990.

[19] Andy Clark and David J. Chalmers. The extended mind. *Analysis*, 58(1):7–19, 1998.

[20] Stéphane Conversy, Stéphane Chatty, and Christophe Hurter. Visual scanning as a reference framework for interactive representation design. *Information Visualization*, 10(3):196–211, 2011.

[21] B. Jack Copeland. Hypercomputation. *Minds and Machines*, 12(4):461–502, 2002.

[22] Anke Dittmar and Peter Forbrig. A unified description formalism for complex hci-systems. In *Third IEEE International Conference on Software Engineering and Formal Methods (SEFM 2005)y*, pages 342–351, 2005.

[23] Paul Dourish. *Where the Action Is*. MIT Press, 2001.

[24] Fred Dretske. *Naturalizing the Mind*. MIT Press, 1995.

[25] Jean-Pierre Dupuy. *Aux origines des sciences cognitives*. La Découverte, 1999.

[26] Douglas Engelbart. Augmenting human intellect: A conceptual framework. Technical Report AFOSR-3223, Stanford Research Institute, Menlo Park, CA, 1962.

[27] Paul M. Fitts. Human engineering for an effective air navigation and traffic control system. Technical report, Ohio State University Research Foundation Report, Colombus, OH, 1951.

[28] Jerry Fodor. *Representations*. MIT Press, Cambridge, Massachusetts, 1981.

[29] Leo Frishberg. Presumptive design, or cutting the looking-glass cake. *Interactions*, 13(1):18–20, January 2006.

[30] Howard Gardner. *The mind's new science*. BasicBooks, 1985.

[31] Saul Greenberg and Harold Thimbleby. The weak science of human-computer interaction. In *CHI'92 Research Symposium on Human Computer Interaction*, Monterey, California, May 1992.

[32] Yves Guiard and Michel Beaudouin-Lafon. *International Journal of Human-Computer Studies. Special issue: Fitts law 50 years later: Applications and contributions from human-computer interaction*, volume 61(6). Academic Press, Inc., 2004.

[33] Steve Heims. *The cybernetics group*. MIT Press, 1991.

[34] T. Hewett et al. *ACM SIGCHI Curricula for Human-Computer Interaction*. ACM Press, New York, 1992.

[35] M. Hirsh. The dynamical system approach to differential equations. *Bulletin of the American Mathematical Society*, 11(1):1–64, 1984.

[36] C. A. R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8):666–677, 1978.

[37] Kristina Höök and Jonas Löwgren. Strong concepts: Intermediate-level knowledge in interaction design research. *ACM Trans. Comput.-Hum. Interact.*, 19(3):23:1–23:18, 2012.

[38] Kasper Hornbæk and Antti Oulasvirta. What is interaction? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 5040–5052, New York, NY, USA, 2017. ACM.

[39] Edwin Hutchins. *Cognition in the Wild*. MIT Press, 1995.

[40] Alan C. Kay. The early history of smalltalk. In *History of Programming languages—II*. ACM, 1996.

[41] David Kirsh. The intelligent use of space. *Artificial Intelligence*, 73(1-2):31–68, 1995.

[42] David Kirsh. Embodied cognition and the magical future of interaction design. *ACM Transactions on Computer-Human Interaction*, 20(1), 2013.

[43] V. Kostakos. The big hole in hci research. *Interactions*, 22(2):48–51, 2015.

[44] Meinard Kuhlmann, Holger Lyre, and Andrew Wayne, editors. *Ontological Aspects of Quantum Field Theory*. World Scientific, 2002.

[45] Thomas S. Kuhn. *The Structure of Scientific Revolution*. The University Chicago Press, Chicago, 1962.

[46] John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: An architecture for general intelligence. *Artif. Intell.*, 33(1):1–64, September 1987.

[47] E. A. Lee. Cyber physical systems: Design challenges. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 363–369, 2008.

[48] Joseph Carl Robnett Licklider. Man-computer symbiosis. *Institute of Radio Engineers Transactions on Human Factors Electronics*, HFE-1:4–11, 1960.

[49] John Randolph Lucas. Minds, machines, and Gödel. *Philosophy*, 36:112–137, 1961.

[50] Andreas Lüdtke, Jan-Patrick Osterloh, and Florian Frische. Multi–criteria evaluation of aircraft cockpit systems by model–based simulation of pilot performance. In *Proceedings of ERTS - Embedded Real Time Software and Systems 2012*, 2012.

[51] Wendy E. Mackay. The interactive thread: Exploring methods for multi-disciplinary design. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, DIS '04, New York, NY, USA, 2004. ACM.

[52] David Marr and Tomasio Poggio. From understanding computation to understanding neural circuitry. Technical Report A.I. Memo 357, MIT AI Laboratory, May 1976.

[53] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[54] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review*, 95:340–370, 1988.

[55] Robin Milner. *A Calculus of Communicating Systems*. Springer-Verlag, 1980.

[56] Brad A. Myers. A brief history of human computer interaction technology. *ACM interactions*, 5(2):44–54, 1998.

[57] Allen Newell. Physical symbol systems. *Cognitive Science*, 4:135–183, 1980.

[58] Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: Symbols ans search. *Communications of the ACM*, 19(3):113–126, 1976.

[59] Society of Automotive Engineers. *Taxonomy and Definitions for Terms Related to On-road Motor Vehicle Automated Driving Systems*. 2018.

[60] Roger Penrose. *Shadows of the Mind*. Oxford University Press, 1994.

[61] Robert Port and Tim Van Gelder, editors. *Mind as Motion: Exploration in the Dynamics of Cognition*. MIT Press, 1995.

[62] Hilary Putnam. Minds and machines. In S. Hook, editor, *Dimensions of Mind*. New York University Press, 1960.

[63] Zenon W. Pylyshyn. *Computation and Cognition*. MIT Press, Cambridge, Massachusetts, 1984.

[64] J. Rasmussen. System design for human interaction. chapter Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models, pages 291–300. IEEE Press, Piscataway, NJ, USA, 1987.

[65] Nicholas Rescher. *Process Philosophy: A Survey of Basic Issues*. University of Pittsburgh Press, Pittsburgh, 2000.

[66] Yvonne Rogers. New theoretical approaches for human-computer interaction. *Annual Review of Information Science and Technology*, 38(1):87–143, 2004.

[67] Wesley C. Salmon. *Scientific Explanation and the Causal Structure of the World*. Princeton University Press, 1984.

[68] RA Schmidt, Howard Zelaznik, B Hawkins, James Frank, and J T. Jr. Quinn. Motor-output variability: A theory for the accuracy of rapid motor acts. 47:415–51, 10 1979.

[69] John Searle. Minds, brains and programs. *Behavioral and Brain Sciences*, 3:417–457, 1980.

[70] Johanna Seibt. Forms of emergent interaction in General Process Theory. *Synthese*, 166:479–512, 2009.

[71] Johanna Seibt. Process philosophy. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. 2013.

[72] Thomes B. Sheridan. Task analysis, task allocation and supervisory control. In Martin G. Helander, Thomas K. Landauer, and Prasad V. Prabhu, editors, *Handbook of Human-Computer Interaction*. Elsevier, 1997.

[73] Ben Shneiderman, Stuart Card, Donald A. Norman, Marilyn Tremaine, and M. Mitchell Waldrop. Chi@20: fighting our way from marginality to power. In *CHI '02 extended abstracts on Human factors in computing systems*, pages 688 – 691, New York, NY, USA, 2002. ACM, ACM.

[74] Herbert A. Simon. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, USA, 1969.

[75] Wolf Singer. Neuronal synchrony: A versatile code for the definition of relations? *Neuron*, 24:49–65, 1999.

[76] Erik Stolterman and Lars-Erik Janlert. *Things That Keep Us Busy – the elements of interaction*. MIT Press, 2017.

[77] Lucy. A. Suchman. *Plans and Situated Actions: The problem of human-machine communication*. Cambridge University Press, 1987.

[78] Harold Thimbleby and Jeremy Gow. Engineering interactive systems. chapter Applying Graph Theory to Interaction Design, pages 501–519. Springer-Verlag, Berlin, Heidelberg, 2008.

[79] Evan Thompson. *Mind in Life: Biology, Phenomenology, and the Sciences of Mind*. Harvard University Press, 2007.

[80] James T. Townsend and Jerome Busemeyer. Dynamic representation of decision-making. In Robert Port and Tim Van Gelder, editors, *Mind as Motion: Exploration in the Dynamics of Cognition*. MIT Press, 1995.

[81] A. M. TURING. I.—computing machinery and intelligence. *Mind*, LIX(236):433–460, 1950.

[82] Alan Turing. On computable numbers, with an application to the entscheidungsprobem. *Proceedings of the London Mathematical Society*, 2(42):230–265, 1936.

[83] Tim Van Gelder. It's about time: An overview of the dynamical approach to cognition. In *Mind as Motion: Exploration in the Dynamics of Cognition*. MIT Press, 1995.

[84] Tim Van Gelder. The dynamical hypothesis in cognitive science. *Brain and Behavioral Sciences*, 21:1–14, 1998.

[85] Francisco Varela, Evan Thompson, and Eleanor Rosch. *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, 1991.

[86] Alonso H. Vera and Herbert A. Simon. Situated action: A symbolic interpretation. *Cognitive Science*, 17:7–48, 1993.

[87] Peter Wegner. Why interaction is more powerful than algorithms. *Communications of the ACM*, 40(5):80–91, 1997.

[88] Norbert Wiener. *Cybernetics, or Control and Communication in the Animal and the Machine*. Cambridge: MIT Press, 1948.

[89] R.S. Woodworth. The accuracy of voluntary movement. 3:i–114, 01 1970.

[90] Shumin Zhai, editor. *ACM Transactions on Computer-Human Interaction (TOCHI) - Special Issue on Physiological Computing for Human-Computer Interaction*, volume 21, New York, NY, USA, 2015. ACM.