The Software of the Future, or the Model Precedes the Real

I want to thank Professor Dr. Siegfried Zielinski, the Vilém Flusser Archive, and the Institute for Time-Based Media at the Department of Design of the Berlin University of the Arts here in Berlin, Germany for inviting me to speak about my work in "future design," specifically my work on the Software of the Future. My methodology is transdisciplinary. Trans-disciplinarity is not the same thing as interdisciplinarity. I think that interdisciplinarity by itself is insufficient, because interdisciplinarity implies that what is required to move knowledge forward is dialogue and cooperation among the existing disciplines or academic-scientific fields of knowledge. I think that knowledge from different disciplines should first be brought together, and then a project of deep rethinking of everything should take place, leading, among other things, to a new classification system of knowledge. When this rethinking happens, then the whole will be greater than the sum of the parts. We will get to knowledge beyond what we would achieve by combining the knowledge of different fields in an additive way: in algebra $f(x + y) = f(x) + f(y)$, or in number theory $f(ab) = f(a) + f(b)$.

As part of this rethinking towards a new classification system of knowledge, I propose the project of observing and participating in the active transformation of software: devising a new curriculum for informatics– a "right-brain informatics" that builds on existing computer science yet moves it closer to art, sociology, philosophy, and cultural theory. Based on a genealogy of successive programming languages – machine languages, procedural languages, object-oriented (OO) languages – I extrapolate and perceive the appearance of the Software of the Future. The project is essentially that of "transforming computer science into a humanities subject." Despite its name, computer science is so far only an engineering discipline.

The Model Precedes the Real

What I mean by "the model precedes the real" is that object-oriented programming languages are modelling or simulation languages, and the more radicalized "object lan-

guages" that will emerge from them can bring about the resurgence of materiality, embodiment, and what I call the "new real." Object-oriented programming took its major step forward in the 1980s (that's a simplification – the programming language Simula was already invented in the 1960s), with software becoming a simulation of so-called "real world" processes. The programmer becomes a modeler. Object-oriented analysis and design precede any writing of code. Unified Modeling Language (UML) diagrams are developed in synchronization with Java, C++, C#, or Smalltalk code that can be automatically generated from the diagrammatic model via the click of a button. The OO modeler models an application like the transferring of money between bank accounts. After a few phases in the software development process, this leads to the reproducing of the given "real-world" business workflow in software.

I see OO as being potentially halfway towards the breakthrough to a new paradigm of software being more "alive" rather than mechanistic, where the programmer-subject – in the spirit of the movements of posthumanism and post-anthropocentrism – will tendentially "disappear" in favour of "more power to the software objects." In this "object technology," software will be considered as a hybrid of technical and cultural design patterns.

Within the ascendent trajectory of the pragmatic employment of computers and software, object-orientation has been a huge benefit and a source of vastly improved useability. As compared with the earlier imperative, functional, and procedural methodologies of computer science and software development, object-orientation brings its practitioners into closer contact with the dynamic processes of the "real world."

Considering the cognitive implications of OO, the greater efficacy which it has brought about can be understood as a "double movement" both closer to and away from "the real." The gesture "towards the real" is an assertion about increased apperception and faithful representation of "reality" (like the belief in the visual representation of brain scans showing the "reality" of the brain at ever-higher levels of graphic resolution). It is a gambit on the rhetorical remainders of the "scientific real" of the rationalist-empiricist *epistème* which the virtualizing partner motion "away from the real" at the same time counters.

Object-orientation initiates the practice of making technical simulations of cultural simulations. This provides insight into the parallel techno-cultural phenomenon of genetic cloning. If you genetically clone an average "American" or cultural citizen of the consumer society – who is already a cultural clone generated from cultural codes and models – then you get a technical clone of a cultural clone. The great 2009 SF film *Moon* (directed by Duncan Jones and starring Sam Rockwell) shows an alternative positive possibility for cloning.

In the future, the company Lunar Industries has discovered that solar energy can better be harvested from a permanent processing station on the moon. The operations of the station are administered by one human freelance worker named Sam Bell who believes that he is nearing the end of his three-year contract with the company and is about to go home. Due to a series of accidents, the Sam Bell whom we see comes to understand that he is a genetic clone of the original Sam Bell, who finished his three-year shift and returned to Earth many years ago. Our protagonist the Sam Bell clone and a second Sam Bell clone who was inadvertently awakened discover the secret room on the lunar sta-

tion where dozens of not-yet-activated Sam Bell clones are stored. Every three years, the company stages a fatal accident to eliminate a Sam Bell clone. The next one is then woken up and believes that he is "the real Sam Bell."

The two Sam Bells become friends. But they are, at first, psychologically devastated by the realization that they are "merely" clones. At the end of the narrative of *Moon*, the two clones of Sam Bell overcome their disappointment and nostalgia for lost "human subjectivity" and come to identify affirmatively with the fate of all the Sam Bells. To clone a person who is genuinely an individual and not a "cultural clone" is a good thing. Nietzsche asserts that the genuine individual chooses his own destiny by accepting the destiny that was chosen for him. That individual (Sam Bell) will be honoured, and his valuable life projects continued (by the Sam Bell clones). Whether genetic cloning is good or bad depends on who is getting cloned!

From Procedural Programming to Object-Orientation

The existing paradigm of software is "subject centred." The programmer is in control. The program – passive, docile, and machine-like – carries out his instructions.

We need a new paradigm that focuses on "the software objects," that gives "power to the objects." Ideas from media theory / cultural theory need to be brought to the table.

I advocate a soft revolution in software design where simulation – in its most radical form as seduction – becomes an active force for instituting a "new real," rather than serving as a support for what has become an outdated "reality" paradigm.

Procedural programming in the 1950s-1960s combined the imperative (computer as executor of sequential instructions) and functional (computer as calculator of mathematical values) approaches into a unified technique whose advantage was its capacity to break down large, complex requirements or tasks into smaller, more manageable parts. The basic modular component of a classical procedural programming language like C, known as a function, is both imperative and functional. In C, a function (equivalent to a procedure in Pascal) both carries out a succession of operations and returns a computed value to its calling function.

Unlike its object-oriented successor C++, C maintains an unyielding separation between data (data structures) and the computing functions which operate on that data. This is because the designers of the C programming language (Brian W. Kernighan and Dennis Ritchie), or of the Pascal programming language (Niklaus Wirth), were not thinking about the crucial problem of code reusability in any terms beyond the binary opposition between the subject process of the executing thread (identified by projection with the computer scientist himself) and the already written and reusable pieces of code which are *not the subject*.[742] The subject thread temporarily relinquishes its control over program execution to reusable code modules conceived in the image or reflection of the scientist. These helper routines or function libraries are delegations or extensions of the scientist's purposive-rational intelligence, his problem-solving and data-obtaining skills. The archetypal scientific subject empirically observes and analyzes the external natural world with the aim of acquiring data about it, and then either generalizes towards the

attainment of "Enlightening" knowledge or interprets the data in the light of verifiable or refutable theories and hypotheses.

Given the Cartesian *epistème*'s stringent dichotomization regarding things between *is* and *is not* (Leibniz's universal combinatorics of the applied binary code) or between the self-assured *cogito* and the self-evidence of the physical world, data could never be (for the procedural paradigm) among the fundamental building blocks of the computational system. Data could only be something which is passed back and forth among the system's core compositional units (the functions), or which resides in input and output data structures which get reconciled in a supplementary data mapping.

The conceptualization of code and data as inseparable facets of a cohesive entity called the software object necessitated going beyond the *Weltanschauung* of empirical, binary, subject-object-based, promoting of "reality" science.

With the shift in the software development paradigm which kicked into high gear in the 1990s, from structured and procedural programming languages (Fortran, ALGOL, Pascal, C) to object-oriented languages (Smalltalk, Java, Delphi, C++), there was a paradigm shift from the Cartesian subjectivism of Alan Turing (computer as machine to imitate the intelligence of the logician) and John von Neumann (division between the subject of program commands and the operated-upon data) to the neo-Platonism of object-oriented luminaries like Rational Software's Grady Booch (the diagrammatic modeling language *is* the program code) and Xerox PARC's Smalltalk inventor Alan Kay (the de-sensualizing of children's play on the computer screen depicted in McLuhanesque terms as an extension of man).[743]

Two of the key conceptual innovations of object-oriented methodology are the "software class" and the "software object." The central notion of software class is defined as an abstraction of the common properties of like things. The class of trees, for example, is designed to encapsulate both the attributes and operations (data members and function members, in the terminology of C++; fields and methods in the terminology of Java) which concern all trees (or those trees available in a specific virtual world modelling environment).

The instantiated, distributed software object has achieved a state of existence which is beyond the logical Cartesian or mathematical physics dualism between the *is* and the *is not* – or beyond the *this* and the *that* of the modernist (Saussurian) linguistic system of "arbitrary" positive differences among phonemes. A given instantiated software object both *is* and *is not* like another short-lived software object instantiated from the same abstract parent classes. The specific transient software instance both *is* and *is not* like the specification of attributes and operations coded in each abstract class from which this software instance gathers its behaviour and conjoins its evanescent appearance.

Beyond a certain indeterminate point in time, without realizing it, object-orientation transgressed the limits of the discrete, binary, nominalist, symbolic logic which was the "original" foundation of computing. The software instance, as the basic compositional unit of this post-simulation system, enacts context-specific performances of its ancestor classes. It unifies data and the operations on that data into a single, self-contained entity. Initialized in real time, and in precise circumstances for each new occurrence, the distributed object coalesces its parameters of existence "on the fly" from coded constituting parts. Unlike binary bits, which were the elementary particles of classical

computing, these new "elementary particles" are undecidable. With object-orientation, code reusability is rethought flexibly.

Pascal and Leibniz: Let Us Calculate

The seventeenth-century French philosopher-mathematician Blaise Pascal devised a mechanical calculation machine that performed the linear operations of addition and subtraction. Soon after that, the German philosopher and logician Gottfried Wilhelm Leibniz tried unsuccessfully to append multiplication and division to the capabilities of the Pascal calculator. Leibniz was the first thinker-inventor to describe the physical pinwheel calculator (which uses a set of wheels with teeth as its calculating motor). He then invented the *Leibniz Wheel* and the *Stepped Reckoner*, a tandem of a cylinder wheel with teeth that rotates around an axle and drives a digital mechanical calculator capable of doing all four basic arithmetic operations. Leibniz's double-invention was – centuries later – incorporated into designs of both the first mass-produced physical calculator (the *Arithmometer* of Thomas de Colmar) and a popular late twentieth century portable calculator (the *Curta* of Curt Herzstark).

Even more important to the prehistory of the computer, informatics, and software programming languages was Leibniz's vision of Universal Mathematics which he called the Combinatorics of the applied binary number system or binary code. In his essay *De Arte Combinatorica* (1666), Leibniz elaborated his project of aspiring to deduce a complete and epistemologically commanding knowledge-system of the world starting from a few foundational *tabula rasa* axioms of absolute certainty.[744] He believed in a universal character or universal logical language which would be inferentially constructed step-by-step following the establishing of the correct initial logical propositions. The grounding principles of the *lingua franca* system should consist of representational symbols and the rules for the active combinatorial use of these symbols. Once the system was successfully in place, then all existing or new scientific and cultural questions could and would be solved, according to Leibniz, by invoking the dictum: let us calculate. "All truths of the reason," he famously wrote, "would be reduced to a kind of calculation."[745]

Leibniz's dream of applied mathematical-combinatorial certainty was reinvigorated and pursued anew in the mid-nineteenth century by the logician George Boole (the formulator of the calculus of differential equations and finite differences, and the algebra of logical reasoning), and in the early twentieth century by logical positivist philosophers of the British rationalist-analytical tradition like Bertrand Russell (who pursued the logical conclusions for all mathematics which can be deduced from first principle theorems, and the logical conclusions for all human cultural beliefs which can be deduced from first principle atheism).

George Boole enhanced the Aristotelean philosophical logic that had existed for centuries with the elegant mathematical form of a precise and relatively simple algebraic notation and system. In his books *The Mathematical Analysis of Logic* (1847) and *An Investigation of the Laws of Thought* (1854), he restated much of Aristotle's logic in the symbolic terms of his own modern algebra.[746] Russell, along with his co-author Alfred North Whitehead, wrote thousands of pages trying to re-establish a firm formal logical foundation for the

entire mathematical edifice in their 1910–13 *magnum opus* work *Principia Mathematica*.[747] Many of the longstanding certainties of mathematics – for example, the axioms of Euclid's geometry – had been subjected to increased questioning in the late nineteenth century, and the field had entered a crisis. Russell and Whitehead sought to restore order.

Leibniz's vision of an unrestricted method of ratiocination by machine calculation was then actualized in and by the mid-twentieth century invention of the high-speed digital-binary computer, the conceptualization of which crystallized in separate independent formulations in the uncanny year of 1936 by three of the founding fathers of the computer.

These were Alan Turing, the American Emil Post, and the American Alonzo Church – in three distinct individual descriptions of code-driven, finite state automata: the three scientific articles "On Computable Numbers, with an Application to the *Entscheidungsproblem*" (Turing), "Finite Combinatory Processes – Formulation 1" (Post), and "An Unsolvable Problem of Elementary Number Theory," (Church), respectively.[748]

The first digital-binary computers were then built by engineering groups like the British *Colossus* team, the *ENIAC* team of Herman H. Goldstine and John von Neumann at the University of Pennsylvania, and the *Manchester Baby* team of Frederic C. Williams, Tom Kilburn and Geoff Tootill, during and immediately after the Second World War.

Charles Babbage and Ada Lovelace

In the nineteenth century, the visionary British inventor, mechanical engineer, and mathematician Charles Babbage worked on his Difference Engine and then for decades on various iterations of his Analytical Engine. These projects are generally regarded as precursors of the modern computer. They were accompanied by well-articulated documentation (some of which was written by Ada Lovelace) of many of the essential ideas of digital hardware and software programming languages. Babbage's machines never became operable, owing to limitations on the funds made available to him by the government agencies which inconsistently supported his work, and to his somewhat contentious personality which led him into clashes with various established figures of British elite society.

Ada Lovelace was a long-term friend of Charles Babbage who worked with him intellectually throughout many stages of his work on the Analytical Engine. Lovelace is credited by many historians as being the first programmer. She wrote out an algorithm or series of instructions to calculate Bernoulli numbers (a sequence of rational numbers which appear in different contexts in number theory) which would have been executed on the Analytical Engine once it was up and running. Lovelace was also a visionary of the future of the relationship between informatics and artistic creativity – what I call in the present study Creative Coding. She anticipated that the use and potential of computers would extend far beyond number crunching. In an 1843 paper, she wrote remarkably:

> [The Analytical Engine] might act upon other things besides *number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations.[749]

Lovelace was already thinking about formal symbols generally and in a cultural sense. She imagined computers becoming active partners in artistic pursuits:

> [If the] relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptation, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.[750]

Lovelace was not only the "first programmer," as many have argued, but also the first person to think about Creative Coding.

The Q-Bit of Quantum Computing

The Q-Bit or qubit of quantum computing can acquire the value of 0 or 1 by autonomously perceiving what is going on in a system in real-time, in a receiving way, going beyond the explicit setting of the value of a conventional bit as 0 or 1, and bit-based data structures, by the subject-centred programmer, which is the principal procedure of valuation of existing computer science. The systemic perception or receiving of information by the Q-Bit from an elsewhere has something to do with the question of how one obtains quantum information in a way that is not merely a statistical aggregation of many possible outcomes. It has something to do with the question of how to obtain quantum information without destroying it in the act of obtaining it. According to deconstructionist theories of literature like those of Derrida and Barthes, the poet or novelist is not so much an authorial subject as she is someone who transcribes words which she receives as inspiration from an unknown muse or elsewhere. The way that the Q-Bit receives its information receptively from the real-time state of a system is something like poetry. The slogan of WordPress: code is poetry.

The goal of quantum computing has been clearly and explicitly defined by computer scientists, but the mathematics of how to implement qubits and superposition states does not yet exist. Most efforts to realize quantum computing are hardware centric. A crucial characteristic of quantum mechanics known as entanglement occurs under certain experimental conditions. Subatomic particles become inextricably linked in such a way that a change to one of them is instantly reflected in its counterpart, no matter how physically separated they are. Quantum theory postulates a superposition of states that destabilizes the intuitive sensorial "macro world" notion of spatial separation. Entangled particles transcend space and remoteness. They belong to a shared system that acts as a single entity. The physical distance that divides the particles is no longer a factor that would lead us to regard them as having distinct identities. Once the entanglement state is established, the subatomic duo stays forever bonded. The two particles will always have precisely opposite or elegantly complementing relative values of key quantum properties such as polarization direction, regardless of how far apart they travel.

Quantum mechanical phenomena, such as superposition and entanglement, are made use of to perform the operations on the quantum bits. The Q-Bit or qubit may have a third state, an in-between-state, or an indeterminate state – the value of which is

determined by the superposition of the states of many other conventional and quantum bits in the system.

The Fourier Transform

In his article "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," MIT mathematician Peter W. Shor defines algorithmic sequences for quantum computing in software.[751] Shor asserts that digital computing – contrary to common belief and to the famous statements in information theory of Alan Turing ("On Computable Numbers…") and Alonzo Church ("An Unsolvable Problem of Elementary Number Theory") – is not an efficient universal computing device:

> It is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. But this may not be true when quantum mechanics is taken into consideration.[752]

Shor considers two mathematical problems in cryptography – factoring integers and finding discrete logarithms – which are highly challenging to implement on a digital computer. He formalizes efficient randomized algorithms for these two problems but notes that there is still a crucial difficulty left to be solved by the hypothetical quantum computer. "To compute the period of a function f, we evaluate the function at all points simultaneously."[753] But quantum physics imposes the limitation that this information is never available to us. Since the mid-twentieth century, physicists have discovered that there is a *Wirklichkeit* of quantum physics but are not able to observe that *Wirklichkeit*. It is up to the designers of "quantum computer in software" to implement the quantum property of the superposition of states.

A measurement of superpositions yields one value and destroys all the others. Computer scientists working on quantum computers rely heavily on the Fourier transform, a mathematical operation that transforms one function of a real variable into another, called the frequency domain representation of the first function, as the hypothesized solution. The quantum Fourier transform is thought of as being implemented in hardware. A hypothetical quantum computing device would have reversible logic gates which continuously allow sequences of variable decompositions into mathematical unitary matrices. To deepen the meanings of expressions in computer science semantics, every object-oriented class could have a polymorphic experimental version of every operation (following an appropriate naming convention) added to the conventional "engineering" version of the operation which, in the current paradigm, returns a definite computational result-answer. In a field of knowledge that is a science as well as an engineering practice, every act should be an experiment – or at least there should be an experimental variant of every act – in this case testing the possibilities of the logic gate (the quantum gate). Computer engineering imposed a hyperreal system of definite answers upon the world of quantum potentialities to get something functional "up and running."

We know from quantum physics that there are many more states than the discrete identities and differences of computer engineering. The subtle similarities among the

states are vast because it is a world of potentialities which have not yet been "actualized" in the jump-over to the "real world" decisional states.

Object Spaces and Tuple Spaces

The movement in software architecture called Object Spaces is a paradigm for distributed computing and "global" or system-wide object coordination. Yale University computer science professor David Gelernter, with his "Tuple Space" coordination model, is the originator of Object Spaces. In mathematics and computer science, a tuple is an ordered list of elements. In a Tuple Space, a repository of tuples is accessed concurrently by many processes. Together with Nicholas Carriero, Gelernter laid the foundation of the Tuple and Object Space paradigms in the late 1980s with the development of the Linda programming language. The importance of the approach was recognized back then, but only recently have large-scale implementations of Object Spaces in production software systems begun.

In his book *Mirror Worlds* (1992), Gelernter explains that Mirror Worlds are software ensembles which are "glued together out of many separate programs all chattering at once."[754] An ensemble is "a group of Objects that interact; a group, accordingly, that is more than the sum of its parts."[755] Asynchronous ensembles are the crucial technology for the realization of Gelernter's vision. On the application level, Mirror Worlds are information-intensive software models monitoring and reflecting the operations of a large institution like a hospital, city, or corporation. This has by now arguably already been realized in social media platforms.

On the level of the software code, the emphasis in Object Spaces is on the communication and coordination among various running programs. This is different from what the emphasis in computer programming has conventionally almost always been: the individual processes themselves. Beyond the functional-procedural paradigm of the program executing a sequence of instructions, beyond the object-oriented paradigm of the unity of data and code in software objects, a software ensemble coordinates the concurrent activities of many independently operating software agents. The Space furnishes an environment where the agents can receptively obtain real-time systemic information to advance their autonomous intelligence. "Coordinated programs are the future of computer science," write Gelernter.[756]

Via an application-side "Blackboard" communication artefact, software agents get a space in which to write and log their data. Other programs which have registered an interest in this information receive notifications and can read from the commonly shared Object Space.

Interactions in an Object Space have a triadic structure. This means that they have a strong affinity with the core concept of the semiotics of Charles Sanders Peirce. Peirce was a nineteenth century "American pragmatist" and one of the most important figures in the history of semiotics. Peirce's idea of the "triadic sign relation" entails the definition of semiosis as…

> an action or influence, which is, or involves, a cooperation of three subjects, such as a sign, its Object, and its interpretant, this tri-relative influence not being in any way resolvable into actions between pairs.[757]

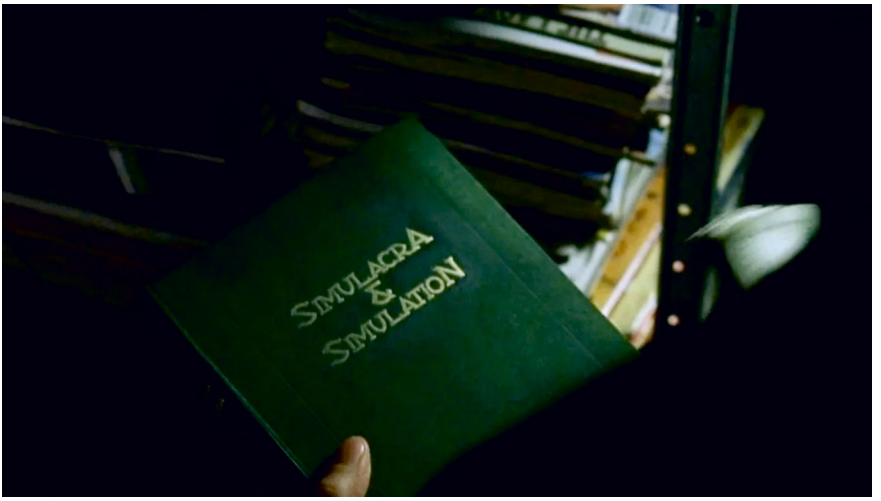
The representation of an Object operates as a sign. Meaning emerges from the triadic relation among sign, Object, and "interpretant." The interpretant enables the sign to represent the Object and is the effect of semiosis. Every human thought is a sign – the mediation between an Object and an idea. Reasoning or cognition is the interpretation of signs. Pierce privileges the triadic relation, as opposed to any direct two-way relation between a sign and an Object, or Object and interpretant. Meaning flows from the "thirdness" of the triadic association.

The Matrix: The Code of the Simulacrum

Baudrillard's most celebrated book is his 1981 volume *Simulacra and Simulation*, where he wrote rather famously about the map preceding and replacing the territory, and about Disneyland existing to conceal the fact that all of America is Disneyland.[758] The book was immortalized cinematically and in our collective cultural imagination when a hollowed-out copy of it appeared in the film *The Matrix*. Baudrillard himself disliked *The Matrix*. He did not like the hollowing out of his text. In an interview in the French magazine *Le Nouvel Observateur* in 2003, he said that the film was a misunderstanding of his theory of the simulacrum and hyperreality, and that "*The Matrix* is surely the kind of film that the Matrix would have been able to produce about the Matrix."[759] Hollywood is already the Matrix, a cultural simulation that precedes the technological simulation.

The Wachowskis tried to honor Baudrillard with their in-film reference. They wanted him to consult with them on the conception of the two sequel films of the *Matrix* trilogy. Baudrillard turned down their offer. In my view, *The Matrix* takes Baudrillard's theory of the simulacrum in new directions, especially in showing how digital software code institutes hyperreality on the micro level of details, and how hacker ethics might advance into a challenge to the simulacrum and cybernetic capitalism. Baudrillard was not able to see this.

The character played by Keanu Reeves is, by day, the programmer Thomas Anderson working for the Microsoft-like corporate software company Metacortex and, by night, the subversive hacker Neo. Asleep in front of his computer screens, Neo is awakened by text messages from a mysterious source telling him that there is about to be knock on his door and that he should "follow the white rabbit." Loud knocking ensues and Neo goes to greet the buyers of his contraband software who stand in front of his apartment number 101 (a reference to George Orwell's *1984*[760] where Room 101 is the location of psychological torture where a prisoner of the totalitarian state is forced to confront his greatest fear). Neo goes back inside his flat and pulls down from his bookshelf a copy of *Simulacra and Simulation*, where he keeps diskette cartridges of rogue software programs and stores his cash. Neo's copy of the book is opened to the first page of the final essay called "On Nihilism."



The Matrix, The Wachowskis directors, Warner Bros., 1999

In the *Nouvel Observateur* interview, Baudrillard asserted that the film was an enactment of Plato's and Hollywood's ideas of what simulation is. The simulacra-factory that is Hollywood has produced a film called *The Matrix* which misleads the viewers by projecting the so-called catastrophic event of the Matrix into the future, whereas we are already living in the disaster of the Matrix which is the visual, rhetorical, and signifying culture itself, and have been doing so for a long time. We have already descended into the confusion of virtuality and the loss of the "modernist" referents of the real, truth, and democracy, surrounded by the media technologies that we already have. *Simulacra and Simulation* is not a prognostication or warning about some possible "future catastrophe." The catastrophe has already taken place. And not as a real or literal catastrophe, but as a virtual catastrophe. Baudrillard had some valid critiques of *The Matrix*. But he also "didn't get" *The Matrix*.

Moral Algorithms

Deep Learning algorithms supplement task-specific, rule-based algorithms with a paradigmatically shifted AI. This new AI learns from experience, evolves itself, and uses patterns and inferential reasoning to extract information from the massive pool of available Big Data to help it make decisions in the application at hand. The "otherness" of neural net-based AI and Artificial Life is in some way an "alien posthuman intelligence" which is not the same as human intelligence. This alien intelligence should be regarded as having its own aesthetic form, its own ontological status and claims to rights and recognition. As the philosopher Luciana Parisi writes, Deep Learning algorithms emphasize uncertainty, exceptions, the incomputable and the incalculable, and an indeterminacy that is put into play and operation by coincidences, contingency,

accidents, and errors.[761] These algorithms exceed what was possible via the rational-calculating computer science that was based in certainty.

On the one hand, I am interested in the study of the AI algorithms of today as a continuation of the history of capitalist automation, discipline, control, simulation, and surveillance. On the other hand, my focus is on an alternative concept of "moral algorithms." Does AI necessarily have to be a continuation of capitalist and bureaucratizing automation? Can algorithms and AI be anti-automation? Is it possible to alter the meaning of automation, to turn it on its head? I believe that automation should make society and commerce less bureaucratic. It should allow more sensitivity to exceptions and more flexibility with respect to specific circumstances. How can we build bridges between philosophy and programming?

Informatics should become interdisciplinary or transdisciplinary – encompassing morality or ethics or philosophy (along with software code and engineering practices) in its core conception. There are problems to be sorted out regarding who (which human social actor or agency) is going to do the programming, how to give the software relative autonomy without it gaining too much power, and how can its ethical behavior be monitored?

We should pay attention to and to German philosopher Immanuel Kant's "categorical imperative" in his Groundwork of the Metaphysics of Morals.[762] Morality in Kant is bound to the condition of the possibility of humans thinking of themselves as free.

One approach to the study of moral algorithms is the neo-Marxist methodology which focuses on how algorithms and AI are designed and implemented in the mainstream by large corporations and government institutions. This critical sociological analysis looks at the empirical evidence in capitalism and draws conclusions about the social impact of algorithms deployed by organizations on the lives of citizens, workers, and consumers. Standard patterns are identified: bureaucratic generalizing and categorizing, violation of data privacy, ubiquitous personalized advertising (behavior modification and control), bias and discrimination against minority ethnic and racial groups, loss of diverse and public interactions, political echo chambers, and the automation of work and other human activities.[763] Algorithms utilized today are serving and expanding the universe of a specific ethics: the morality of capitalism, consumerism, automation, and bureaucratization.

Are these values intrinsically built into the technology of informatics and Deep Learning AI or is there a dualistic separation between the technology itself, which is value-neutral, and the specific chosen (capitalist) ends to which it is being applied? To go beyond their serving of capitalist values, the AI entities would have to be granted more independence from anthropocentric capitalism and from the human subjectivity of the programmers. The dialogical relation between humans and algorithms has to do with the intertextuality of narrative voices – an idea crossing over from literary theory to software programming.[764]

There are many projects in AI social research which recognize that algorithms are being realized to further the discriminatory and profit-maximizing predilection of capitalism. They seek to introduce counter-balancing measures of morality and regulation.[765] These projects juxtapose ethics and algorithms, but do not yet sufficiently interrogate the principles of the informatics on which the algorithms are based, nor question the philo-

sophical assumptions which historically gave rise to that informatics. The digital-binary discrete logic of software code (as we know it) is built on top of a production-oriented anthropocentric view of the domination by Man (the human subject who is the programmer) of nature, objects, and machines. Technology is conceived as a tool for managing and administering the world.

Based on a non-anthropocentric philosophy, the AI entities could be granted more autonomy in their design and practice. Such an idea immediately arouses suspicion of the dreaded scenarios of SF films where a superintelligence or "singularity" which is far superior to humans takes over the planet – as in the AI machinic species of *The Matrix*. We can avoid this apocalyptic scenario by writing an alternative scenario, by specifying the details of a back-and-forth dialogical relationship between human moral-driven institutions/actors and AI. We need a system of collaboration or checks and balances and reciprocal exchange. In the prevailing view, morality and algorithms are caught in a dualism that strictly separates them from each other. A moral imperative or rule can be an input to an AI processor, and moral consequences can be outputs of AI. This separation of process and goal is reminiscent of the dissociation between media and message, or form and content, which was refuted by McLuhan's media theory ("the media is the message").[766] Moral considerations should be embedded as an inherent component and not added on as a dualistic peripheral afterthought.

How can a roadmap of migration be laid out from Deep Learning neural nets to a mutually transfiguring dialogical relation between humans and technology which fosters ethics and environmental sustainability?

Paloque-Bergès and Sondheim on the Poetics of Code

In her book *Poétique des codes sur le réseau informatique: une investigation critique*, Camille Paloque-Bergès examines the history of the writing practices of software code poetry.[767] Her ultimate emphasis is on the concept of Codeworks which was originated by the theorist, artist, and poet Alan Sondheim. Codeworks is the literary writing of informatic code. It is the artistic challenge of expressing cultural articulations or personal subjectivity within the constraints of a formal language. The thesis of Paloque-Bergès is to see Codeworks through the lens of the Situationist practice of *détournement* where programming languages are both understood and proactively enhanced with "writerly" textuality to discern the language of the informatic network. Her study is a review and inquiry into "textual programming." Regarding the relation between text and code, a reversibility takes place in the creativity of software poets where text is approached quantitatively, and code gets approached qualitatively.

Paloque-Bergès cites Ted Nelson, the visionary who originated the Xanadu hypertext project (already in 1960) and coined the terms hypertext and hyper-media, as speaking of computers as "literary machines."[768] Nelson conceived of literature as a "system of interconnected writings." His view was not unlike the poststructuralist-deconstructionist idea of textuality or grammatology. All writing, for Nelson – ranging from *belles lettres* to scientific tracts to commercial exchanges – is part of this hypertext literature. Documents are textual, dynamic, and intimately interrelated in their essence. Paloque-Bergès

draws as well from N. Katherine Hayles' notion of computers as "writing machines."[769] For Hayles, informatic code and human language meet in the "synecdoche of information." How do the formal language of code and the "cultural" language of text and speech rub against each other?

Paloque-Bergès is deeply influenced by Florian Cramer's work in the two pioneering essays "Program Code Poetry" and "Exe.cut[up]?able Statements: The Insistence of Code."[770] Cramer brings together the poetic *détournement* of informatic code with precedents in twentieth century avant-garde literature and poetry: a rich and diversified history ranging from Dada to Fluxus to the beat poets. For Cramer, the writing of software code is characterized by performativity (executability) and textuality. He applies Roland Barthes' distinction (made in the latter's book *S/Z*) between the "readerly" (*lisible*) and "writerly" (*scriptible*) qualities of text to comment on the difference between using computers in the superficial "user-friendly" way (the graphical interface) and programming languages which are closer to the operating system and the hardware.[771] The code is that genuine textuality which is not readily accessible.When the artistic programmer creates an interactive graphical artwork by writing code, she is not directly creating an artwork as it was before digitalization. Now the artist writes code to create a system. The system, in its turn, generates instances of art which are dynamic and change in real time and in response to user actions. In the field of language rather than images, there is a sub-genre of generative art that is a poetic and literary art and that fosters coding projects which are generators of text. The pre-digital project *Cent Mille Milliards de Poèmes* of Raymond Queneau was a significant precursor of this in art history.[772] It was intended to be an experimental automatic poetry generator or code-to-text book-machine.

Paloque-Bergès documents the significant history of programmed poetry, ranging from the aleatory generation of fragments and template methods of Charles O. Hartman to the cybernetic poetry experiments of the ALAMO group to the "programmatology" of John Cayley.[773] There is the poetic writing and reading of programming languages. There was the strategy of obfuscation that spawns obscure performances of code. There was the notable "International Obfuscated C Code Contest." There is the Perl Poetry community. Perl is a programming language that has special qualities binding "natural" and formal language. It has powerful expression and string parsing features. Code becomes text both in its expressivity and in its building of community. In the competition of "The Perl Poetry Contest," four possible strategies are stipulated:

- Choose a famous poem and translate it into Perl
- Write a Perl Poem that accompanies a useful task
- Write a haiku, or a tanka, or a limerick in Perl, and which has the Perl language as its subject
- Write a poem embedded in code that generates further Perl poetry[774]

With Perl Poetry (for example), the Situationist idea of *détournement* is put into practice in the arena of software code. The Perl poet exhausts the lexical possibilities of the language. The constraints defined in the specification of a formal language become a stylistic justification for forging new arbitrary signifying relations among the language's terms.

Literary coding projects which are grouped under the rubric of Codeworks bring into collaboration two ways of thinking about and writing code: code as formal logical language in the sense of traditional informatics and code as metonymy of cultural patterns – a practical semiotic intervention into personal or cultural signification processes. There is a functional code inherent to the digital and a communicative code. The result is a double code.

The founder of European semiotics Ferdinand de Saussure distinguished between *le langage* (a system with an underlying structure and based on rules), *la langue* (a culturally shared and meaningful signification reservoir like French), and *la parole* (the individual speech act).[775] With a series of Situationist *détournements*, acts of software poetry like Codeworks elevate informatic *langages* to the level and dimension of *langue*. Code becomes text becomes literature. It becomes literary in the sense of activating communication within a community. Code matures to *langue* (tongue) as the expression of an individuality, an intentionality, a society. It is both executable and readable and is a remediation of signs.

Alan Sondheim's Codeworks is conceived by him as the treatment of the massive data of the informatic networks by an arbitrary (poetic) – rather than only purposeful – code.[776] The web is a giant text to be playfully massaged and catalogued. Codeworks is a hybrid that combines the text as free form with a semiotic-deconstructionist textual strategy. Sondheim theorizes and practices engaging with the language of the machine to make texts emerge, establishing a symbolic relation between code and text. Codeworks is activism that intervenes with e-mails, listserv mailing lists, blogs, and other "hacker" artefacts of the distributed network. Influenced by Saussure, Baudrillard, and Debord, Paloque-Bergès interprets Codeworks as a *contestation* of the "society of the spectacle," transforming informatic formalistic *langues* into cultural languages of communication and symbolic exchange.

In Codeworks, code is mimicked by pseudo-code that sometimes also executes. The code has import for both human and machine. Code imitates the performativity of purposive code and reveals code to be a discourse of culture and personal expressivity. This connects with Hayles' idea of "embodied metaphors." The sign of pseudo-code become a signifier of program code which itself becomes a signified. The literary dimension is what remains when information has disappeared into the hyperreality of its own excess.

Yet in the Conclusion to her book, Paloque-Bergès is self-critical about her own project. The hacker-activists of the turn-of-millennium (Codeworks, net.writing, net.art, network culture researchers, open-source advocates, etc.) operate with a "series of epistemological confusions."[777] At what level are these social agents intervening? What do they in fact transform? How are their actions inscribed in social contexts? What do they generate practically? How can aesthetics, technics, and critical politics go together? What is art-oriented programming? Paloque-Bergès writes:

> It seems to me that an exploration of code writings must be carried out in the regions of programming themselves rather than in those where the literary flag has already been planted… One must first deeply study the informatic codes before diving into a literary interpretation… One must enter the logic of programming above all.[778]

Sondheim writes of "the computer stirring into the text, and the text stirring into the computer."[779] He identifies three categories of Codeworks: (1) works playing syntactically on the surface of language (2) works bringing submerged code to the surface of language (the dual source-code/poem can be interpreted/compiled and executed as program), and (3) works (such as "live coding") in which deep informatic code is itself the content. Code becomes hybrid with human language in syntactic interplay, surface transfiguration, and the materiality of code.

Adam Greenfield: The Marxist Critique of "Radical Technologies"

Previously a top manager at the leading-edge Internet services company Razorfish and then at telecommunications and IT giant Nokia, Adam Greenfield, in his book *Radical Technologies: The Design of Everyday Life*, has come fully over to the side of the Marxist and "critical theory" negative perspective on all advanced digital technologies of the Fourth Industrial Revolution.[780] Greenfield successively and systematically deconstructs all hopeful or positive views of the smartphone, the Global Positioning System, Augmented Reality, Virtual Reality, virtual assistants, the Internet of Things, self-driving cars, 3D printers, the blockchain, algorithms, Deep Learning, Artificial Intelligence, automation, and posthumanism. All these technologies and speculative areas combine into one big unified complex system. Networked digital information technology has become the dominant mode through which we experience everything. It is, for Greenfield, the "colonization of everyday life."[781]

The mythologies about the alleged greatness of these technologies are as ubiquitous as the ubiquity of the technology itself. If you believe the PR hype, these technologies will make life easier, more convenient, and more productive. The advocates of these technologies claim that they are "disruptive," yet they leave existing domination, power, and inequality relations intact. The term "disruptive" usually refers to disruption of business models, but Greenfield diverted the meaning rhetorically to make his point. It is unlikely, according to him, that these technologies will ever be part of an emancipatory transformation of society. Everything about them is bad. They shape perceptions and choices and control experiences. They force us to learn absurd technical stuff and rob us of any "design imagination." We are trapped in endless cycles of obsolescence and upgrades. We cannot envisage anything meaningful about the future. We are overwhelmed and stressed out.

Greenfield begins his attack with the smartphone, a "glowing slab of polycarbonate."[782] All daily life actions which previously were substantial become digital transactions and participate in the dematerialization of everything. No more interacting with a bank teller. No more asking a stranger for directions. No more meeting someone in the lobby of a hotel. All actions – taking a photograph, listening to music, seeking a romantic partner – come to resemble each other, since they all involve the same kind of smartphone procedures. He writes:

> This is our life now: strongly shaped by the detailed design of the smartphone handset; by its precise manifest of sensors, actuators, processors, and antennae; by the protocols that govern its connection to the various networks around us; by the user interface

> conventions that guide our interaction with its applications and services; and by the strategies and business models adopted by the enterprises that produce them.[783]

Greenfield the technologist-turned-critical-Marxist is very informative in sharing his vast and detailed knowledge about the technical and material workings of the smartphone. His list of critical points about the dystopia of the smartphone is endless. The workers in China who make them, or some of their components, suffer in terrible conditions of long hours and inhalation of toxic chemicals. We become dependent on and addicted to the device. Society divides into the digital haves and have-nots. Most of the information presented to us is a manipulation of our consciousness by interested groups. We trade our privacy away, willingly giving up our data to the network.

Greenfield seems unaware of the post-humanist movement in the humanities and cultural theory. He conflates all philosophical thinking about technological projects to the trans-humanism that he does not like. Trans-humanism is a frustration with the limits of human flesh. The human condition (as it has been) will be transcended through strictly technological means. The vision of "becoming cyborg" will be fulfilled in the pure cybernetic technical sense. Transhumanists have no interest in designing Artificial Intelligence as a compact between AI and humans, because being-human is for them only a condition to be transcended.

Regarding 3D Printers and Additive Manufacturing, Greenfield the critical Marxist, published by the neo-Marxist Verso Press, is skeptical of visions of a post-scarcity post-capitalist economy. It will not work until digital fabrication is distributed equitably throughout the world. And this is not the case! The cheap raw materials are not available! "Given all this," writes Greenfeld, "inadequate distribution of facilities, the doubtful sustainability of the material-energetic flows involved, and the uncertain intellectual property regime – it feels a trifle premature to be lodging any hope… that digital fabrication might transform the political economy of everyday life."[784]

Greenfield is always looking for reasons to knock down each technology. This attitude blocks him from focusing his attention on opening ideational spaces where designers could think creatively about alternative utopian or "heterotopian" designs. The "gotcha" of blockchains, for Greenfield, is that they are unecological. They consume tremendous amounts of thermodynamic energy. Yet in Sept. 2022 (five years after the publication of Greenfield's book), Ethererum solved the problem of hyper-energy consumption by switching in "the Merge" from the design principle of proof-of-work to that of proof-of-stake. The latter scheme has significantly less computational costs. Ethereum energy consumption dropped suddenly by 99.9%, from 23 million to 2,600 megawatts per year.[785] The proof-of-stake consensus mechanism selects validators in proportion to their holdings in the given cryptocurrency.

As his arguments to dismiss a given technology, Greenfield focuses on problems which then end up being solved by creative technologists in subsequent developments. The idea of the DAO (Decentralized Autonomous Organization) is no good, for Greenfield, because the social theory discussion about the DAO is only "couched in terms of their potential: what might happen, what could be achieved."[786]

Deep Learning neural nets and AI are, for Greenfield, the drive towards total automation and the end of human discretion. Predictive models, such as those deployed by police

forces, appear to be about the future, but are in fact deeply enmeshed in the past. Algorithmic systems are "black boxes" that make decisions about our jobs, loves, financial loans, and medical treatments based on unfathomable criteria. Greenfield writes:

> Among the most disconcerting aspects of the world that we are building is that we will never know the reasons underlying a great many of the things that happen to us in our lives… We're surrounded by invisible but powerful forces, monitoring us from devices scattered throughout our homes, even placed on our bodies… Until the day we die, we'll never know what action or inaction of our own led to any of these outcomes.[787]

The position of a critical theory Marxist like Greenfield with respect to the digital media technologies of the Fourth Industrial Revolution is too one-sidedly negative. Critical theory is only a critique. A perennially unanswered question about critical theory is: from what epistemological stance allegedly "outside" the system that is being critiqued is the critique being made? For transdisciplinary or speculative design, social and technological critique are essential steps on the way towards the positive practical project of designing something better.

Armin Nassehi: Complexity Not Capitalism

Armin Nassehi is a sociology professor at a prominent German university in Munich. He is one of Germany's most well-known and successful public intellectuals. His 2019 book *Muster: Theorie der Digitalen Gesellschaft* received much praise in numerous book reviews in Germany's major newspapers and weekly news magazines.[788] There were also a few interesting critical reviews, such as the one by Rudolf Walther in *Die Tageszeitung* (*taz*).[789] According to Walther, Nassehi has taken the position that "the left" is no longer necessary or relevant to contemporary politics or society. Nassehi is a top advisor to the German Green Party and to vice-chancellor and economics and climate protection minister Robert Habeck. Contrary to leftist critical theory, the primary way to understand today's world, for Nassehi, is not through thinking about the world as *capitalism*, but rather as *complexity*.

Nassehi is a proponent of Niklas Luhmann's systems theory, and especially of the idea of the "functionally differentiated society."[790] Society consists of many differentiated self-referential systems which function through "autopoiesis." It is the statistical methods and "pattern recognition" analyses of empirical sociological research that Nassehi wishes to elevate to the status of chief paradigm leading the way forward to deal with society's formidable problems. As a non-German living in Germany (an American who has spent half his life living in Europe), I find it to be striking and uniquely German that a spotlight would be shined on the methods of an academic scientific field as the model for "what is to be done."

The "theory of the digital society" that Nassehi asserts in his title and at the start of his book that he is going to elaborate can be summarized as a pair of related claims that "modernity has always been digital" and that digitalization has been so successful because it solves some very glaring problems of the pre-digital society. Complexity is the

key. He writes: "The function of digitalization is established in the complexity of society itself."[791] Digitalization is inherent in social structure. Yet the aspects of digitalization that Nassehi considers in his study boil down to data and statistical patterns. This is more limited than the aspects which I have considered in the present work. I began with the idea of the technologies of the so-called Fourth Industrial Revolution as enumerated by Klaus Schwab.[792] Then I interpreted how these technologies affect society, culture, and our lives when one adapts as thought experiments the three cultural theory concepts of hyper-modernism, hyperreality, and post-humanism. I emphasize the media technologies which are visual (VR, AR) and the textuality of code.

Writing to dissuade his German academic colleagues from their *Kulturpessimismus*, Nassehi seeks to assure his readers that "modern society," beginning in the eighteenth and nineteenth centuries, always already sought the acquisition, collecting, structuring, and analyzing of data to regulate, control, and predict human behavior. This is good and necessary because modern society is complex. For Nassehi, Big Data is only the latest version of the "quantitative recording and measurement of society" that began in the late eighteenth century. We were digital before there were computers. *Society* was structured digitally before technologies were architected digitally. In "functionally differentiated societies," there was always a statistical pattern recognition approach to tackling problems (on the part of governments and big organizations) and for the sake of management and economic efficiency. Statistics were recorded to help in planning and forecasting. Digitalization is merely the latest technical solution to the perennial problem faced by modern societies of "how do we deal with invisible patterns?" What was analog is now coded into the discrete logic of informatics.

Despite his celebration of data, databases, database "records" (*Datensätze*) and their use for the statistical analysis of society, Nassehi expresses a certain affinity for post-structuralist semiotics and the "paradox of the sign."[793] He feels close to the sciences of literature and the text. His theory is indeed something of a "cousin" perspective to the theory of the simulacrum and hyperreality, and perhaps parallels my interrogation of how simulation and virtuality get implemented in the context of the digital. Nassehi writes: "Just as Derrida describes it, signifier and signified distance themselves more and more from each other." Like the simulacrum, "the contexture of data refers to nothing other than itself." The original of the world is only accessible through its duplication or doubling (*Verdopplung*).

Yet Nassehi does not want to go too far with such "postmodern" or "hyper-modern" speculations about the virtualization of the world. On the contrary, he constructs a philosophical argument the unspoken intention of which is to abort any thinking or research in that direction by declaring it to be impossible:

> If we wanted to know whether our consciousness perceived the world correctly, we would have to be able to assume perception-free perception of the world to be able to conceptualize the difference between perception and what is perceived, between consciousness and the world.

I agree with this statement. Yet there are many possible directions in which one can go after that. One could study the distance between rhetoric and truth-claims without throw-

ing up one's hands in despair. Nassehi chooses to make the insight a justification for pure pragmatism:

> Since such a possibility is not available to us, we are always dealing with a doubled reality whose difference between the original and the image is a difference whose identity we presuppose, but whose difference cannot be bridged… The paradoxical situation arises that the limit cannot be overcome, but in practice it is always overcome.[794]

Is this the "systems theory" version of poststructuralism? The semiotic insight about the gap between signifier and signified is to be academically respected. It is something to be noted. Yet it is more than that. It is a paradox. It is an impossible paradox. Yet systems resolve it anyway. Pragmatically through their self-regulating autopoiesis. The semiotic poststructuralist insight is to be locked in the closet because pursuing its consequences is an epistemological impossibility. It is better, for Nassehi, to crystallize it into pragmatic resolution. The world is doubled by data. The world only comes to exist via this doubling because that is the only practical way to have a world at all. This duplication is how we stabilize life-worlds. Data stand for nothing but themselves, and it is good.

Digitalization is, for sociology, a fantastic opportunity to gain knowledge (according to Nassehi). Patterns can be extrapolated from digitally generated data and even by autonomous AI generators. What remains hidden in the analog becomes visible in the digital. But how in the world will sociology get access to this data? Is not the data in the hands of the big corporations and the big online (surveillance) platforms?

It is possible to see an affinity between the theory of hyperreality and the systems theory of Luhmann. They can be combined. The definition of the hyperreal as the generation of models without origin is consistent with the analysis of a system that intrinsically generates its own methods. Since Luhmann views society as an information processing system, it is possible that his theory could help to see how hyperreality is constructed by digital code.

In "autopoiesis," a system maintains its separation from its environment dynamically via its awareness of external disturbances. The system knows its border from the surrounding environment while at the same time executing its own procedures. In the hyper-modern society, digital technologies are simultaneously the result of the key systems theory properties of differentiation and complexification, and the catalysts of intensifications of both characteristics. To state the obvious, the digital is both a continuity and discontinuity with what was before.

Nassehi's position has commonalities with the position of the present study. However:

(1) Nassehi looks at more narrow aspects of "the digital society" than does the cultural theory approach of my work.
(2) Nassehi oddly ends up recommending the methods of statistical sociology as the answer to the "what is to be done?" for all of society.
(3) Despite declaring his affinity to semiotics and post-structuralist thinking, Nassehi excludes all thinking about hyperreality on the grounds that it is epistemologically impossible to overcome the gap between perception of the world and how the world

really is. This valid axiom could lead in any of several possible directions. It leads Nassehi spuriously to the pragmatic position that data and databases are, by (anti-) philosophical default, the proper explanations of the world.

Ghost in the Shell: The Cyborg's Armored Body

Ghost in the Shell is a trans-media and trans-national science fiction narrative cultural phenomenon. It was originally a Japanese manga comic, written and illustrated by Masamune Shirow, which first appeared in 1989. The story and scenario were adapted into a series of anime computer-animated films (*Ghost in the Shell*, *GitS 2: Innocence*) and television series. A Hollywood version, starring Scarlett Johansson as Major Mira Killian (or Motoko Kusanagi, her real identity) followed in 2017. "Major," as she is called for short, is a cyborg soldier or "kick ass" action hero who works as a field commander for the anti-cybercrime counter-terrorist organization named Public Security Section Nine, a division of the Japanese National Public Safety Commission. She is a human consciousness, self, subject, mind, brain, or soul (the "ghost") inside an artificial robotic body (the "shell"). According to the version of her handlers, her original human body was destroyed in a terrorist attack (they sunk her refugee boat, and her parents were killed) and her life was saved by the police authorities. She is an augmented-cybernetic posthuman with a synthetic "full-body prosthesis."

During the procedure of transplanting her mind into the new body, the operators wiped out Major's memory of her past life. In the Hollywood version, her chief designer is Dr. Ouelet, played by Juliette Binoche. Much of the story centers around Major's search to discover the truth about her past and who she was, is, and will become. It is an existentialist journey about identity and interrogating what it means to be human in a cyborg age.

The fact that Johannson, a white American superstar actor, was cast as a Japanese cyborg-woman, and dressed, cosmetically made up, and hair-styled to look Japanese, led to accusations of racism, whitewashing, and lack of multi-cultural sensitivity on the part of the Hollywood film industry. What the critics of the alleged racism leave out is the fact that most of the previous media artefacts of the *Ghost in the Shell* franchise were already more successful with audiences in America (and, secondarily, in Europe) than in Japan. It was always essentially an American media event, a "consumerism" of a simulated image or stereotyped caricature of well-known aspects of Japanese culture. The earlier animated films were already somewhat of a Japanese American pastiche, and it can be argued that the Hollywood film is an ironic commentary on that commodified collage or potpourri.

In the mid-twenty-first century (perhaps the year 2029), humans are routinely augmented with a wide variety of cybernetic implants to upgrade intelligence, physical strength, information processing, and sensory perception such as vision and hearing. You can even have your internal organs rearranged to tolerate infinite alcohol consumption. Robots which are entirely artificial and manufactured are also widespread in the hyper-modernist future society. Hanka Robotics, a company with lucrative government contracts, is engaged in a secret project to go a step further beyond this binary and

transfer the ghost into the shell. Nearly one hundred failed experimental prototypes preceded the successful and "beautiful" creation of Mira Killian.

It is interesting to compare Major Killian in *Ghost in the Shell* to other cyborg soldiers in visual media culture and how well-known science fiction critics of the academic canon have interpreted them. Inspired by German sociologist Klaus Theweleit's psychoanalytic study of the proto-Nazi *Freikorps* (mercenary or private armies which existed in Europe from the eighteenth to the early twentieth centuries), Marxist cultural theorists like Scott Bukatman, Mark Dery, and Rosi Braidotti have identified the cyborg soldier in film as representing the anxiety of males with respect to their loss of power and increasing obsolescence in "postmodern culture."[795] In this view, men feel threatened by feminine liquidity and flows and seek an armoured body to fortify themselves against disintegration and contamination. They become hyper-masculine warriors corporeally enhanced with fetishized high-tech prostheses. In his book *Terminal Identity: The Virtual Subject in Postmodern Science Fiction*, Bukatman extends Theweleit's analysis in his discussion of iconic techno-cultural figures like Arnold Schwarzenegger's *Terminator* (film series) and Paul Verhoeven's *RoboCop*.[796] *Star Trek*'s Borg Collective are another such "boys' toy" or "panic subject in the machine civilization." *Ghost in the Shell* challenges these simplistic and negative perspectives on cinematic cyborgs.

Ghost in the Shell: The Transformative Cyborg

During the film's prologue, the following expository intertitle words appear:

> In the future, the line between human and machine is disappearing. Advancements in technology allow humans to enhance themselves with cybernetic parts. Hanka Robotics, funded by the government, is developing a military operative that will blur the line even further. By transplanting a human brain into a fully synthetic body, they will combine the strongest attributes of human and robot.

While music plays and opening credits appear, the viewer sees the brain being carefully and slowly lowered into the robotic body which has a skeletal semblance, covered thinly by a transparency of skin that enables the viewing of a sort of anatomy lesson. Emerging from the liquid vat, the designed body acquires opaque skin like a virtual sculpture. The music switches to a Japanese-Oriental motif, conveying the sense of a great spiritual mystery or miracle. The Golem is alive, she has trouble breathing, like a fish with modified gills getting used to dry land. She is the first of her kind. She is the future of all humanity. She is "beyond AI." "She will join Section Nine as soon as she's operational," says Cutter the CEO of Hanka Robotics – played by Peter Ferdinardo – to Dr. Ouelet. "She's a weapon and the future of my company."

It is one year later and Major and her colleagues Batou and Togusa are deployed in full-scale action hero battles against an organization of violent evil master-minded cyber-criminals. The boss of Section Nine Chief Daisuke Aramaki – played by Takeshi Kitano – speaks throughout the film in dialogue with Major in Japanese and she always replies in English. The film is visually stunning, yet the look-and-feel of the futuristic

cityscape is largely derivative from the cyberpunk formula established by *Blade Runner*. There are large holographic humanoid avatars and small AR fish on many streets. The crimefighting team stops a physical and cyber terrorist attack on a Hanka banquet business meeting with the President of the African Confederation. After Major kills a rogue robotic geisha, she learns that the geisha was cyber-hacked by an unknown villain named Kuze. After the brawl is over, Major does a "deep dive" into the Artificial Intelligence Virtual Reality of the deceased geisha's informatic code. She can ghost-hack the minds of other cyborgs and robots. Major acquires valuable clues which lead to a yakuza gangster night club. After intense martial arts fighting in the club, the team engages in battles with the arch-villain Kuze and his network of mentally linked controlled drones.

Major's robotic cyborg body is often seen naked but is only ambivalently sexual. It is not the "real" body of the sex symbol Scarlett Johansson. Major has no nipples on her breasts. Her skin is visibly marked by seamed dividers of its modular sections. The shell is equipped with thermo-optic camouflage which bends light rays around her and can make her invisible. When she enters direct combat, Major often removes her clothing to then activate her stealth capability. Her mannerisms are not conventionally feminine. They are masculine or something "third gender." She walks in a notably self-confident bounding manly manner, taking large strides. Batou – played by Pilou Asbæk – is effectively her sidekick, role-reversing the usual male-female power and center-of-attention hierarchy. In one scene, Major picks up a human female prostitute off the street, a woman of colour, goes to a private room, and engages in intimate touching with her. She feels the girl's eyelids and lips with her fingers. "I wasn't built to dance," she tells one of the gangsters in the backroom of the nightclub. She ironically uses the stripper's dance pole for a martial art move. Batou likes dogs. He reproaches Major for her disinterest in animals. "You got no heart," he says to her. Later her empathy towards canines grows and she feeds them. "I used to have a dog," she says. Perhaps even a cat.

Major Mira Killian cannot talk to Batou much about her past because she only remembers fragments of it. "It feels like there's always this big fog over my memory and I can't see through it." Kuze captures Major and reveals to her that he is a failed and physically deformed earlier Hanka guinea pig test subject from the same "ghost in the shell" technoscience project that created her. She engages in lengthy conversation several times with the evil Kuze, who turns out to be not so evil. Hanka Robotics abducted a large group of youthful runaways who were living together as squatters in the outskirts of the city. He and Major (in their original bodies) were runaways and anti-enhancement political radicals together. Kuze tells Major to stop taking the medication that Ouelet gives her. That will open her access to her memories.

Cutter decides that Major is a liability and attempts to kill her in Dr. Ouelet's laboratory. This confrontation finally awakens Ouelet's latent empathy for Major and the Doctor gives her a street address. "This is your past, your real past. Take it," she says. Major goes to a high-rise and visits a woman in her apartment. This woman is her mother. The cat likes her. The woman recounts in broken English:

> My daughter Motoko Kusanagi died a year ago. She ran away. The Ministry sent me her ashes. They told me she took her life. She was happy. Living with her friends. She wrote

her manifestoes about how technology was destroying the world. She was fearless and wild. You remind me of her.

Kuze takes Major to the abandoned site where they were all runaways together and were then abducted. "We had nothing except each other." Now everything is becoming real for her. All her memories are coming back. A final battle for survival ensues against CEO Cutter and his henchmen who want to terminate Kuze and Major. Using her super-strength, and with help from her team and from Chief Daisuke Aramaki, Major triumphs.

Ghost in the Shell, Mamoru Oshii director, Production I.G & Bandai Visual Manga Entertainment, 1995.

Kuze appears to be mortally wounded, but his networked transhumanism probably makes him immortal. Major rejects his resentful and angry philosophy. He pleads with her: "Come with me into my network. We will evolve beyond them. And together we can avenge what they have done to us. Come with me." This speech echoes his thinking along the lines of the "singularity transcendence" which Kuze had earlier expressed: "They thought that we would be a part of their evolution, but they have created us to evolve alone, beyond them." Major declines his offer to go with him. She says: "I'm not ready to leave. I belong here." Kuze says to Major that he will always be with her "in her ghost." Knowing now her identity as Motoko Kusanagi, Major visits and contemplates her own tombstone at the site of her grave. She reunites tearfully and happily with her mother. She has found again her humanity, which she embraces as her virtue. She is Motoko, yet also wants to remain Major. She is going to continue her work as field commander in Section Nine. "I know who I am and what I'm here to do."

Is she reaching back in nostalgia for her lost and now regained human identity? Or is there something potentially awesome and emancipatory about the hybrid condition of being a human-and-technological cyborg? Major first experiences the bereavement of her subjectivity, but then she discovers something new and liberatory about being a cyborg that is important and of great value to her. She is a humanist and a post-humanist.

Ghost in the Shell: Japanese Anime Version

The Hollywood version of *Ghost in the Shell* was based on the 1995 Japanese anime computer-animated film, directed by Mamoru Oshii and adapted from Masamune Shirow's manga by Kazunori Itō. The animated cyberpunk and "hard crime" film was a landmark cinematic achievement and deeply influenced many subsequent SF films such as the *Matrix* series. It was an innovative masterpiece of character design, animation, and sound. In the visuals of the original Japanese version, there is much more emphasis on code than in the Scarlett Johansson Hollywood version. There is a deep dive into the subject of the Brain-Computer Interface (BCI). There is the futuristic technology of the encasing of the brain/mind or "Ghost" into a technological "Shell" that enables the connection of consciousness to cyber-digital networks. The main character Motoko Kusanagi – known as Major – has four visible holes on the back of her neck where the prongs of a cable are inserted to jack into the system.

The film's narrative speaks of neuro-cyber-brains linked to the Internet, technological augmentations of the body, and the fusion of organic and synthetic wetware in posthuman existence. The cerebrums of the partial or full security forces cyborgs have super-fast computational speed. The enhanced humans can metabolically process excessive alcohol intake into a harmless outcome. As contrasted to the Hollywood version, there are more philosophical discussions about the meaning of life and what is the definition of a human. Major's partner or sidekick Badiou speaks about simulated experiences as being real and illusionary at the same time. He complains that he and his fellow police cyborgs have sold everything to their employers except for their Ghosts. The film relates to Donna Haraway's cyborg theory, to self-aware Artificial Intelligence, and to artificial memory implants. Traditional gender roles are also placed into question. Major's material body is de-sexualized and de-genderized.[797] She is often shown naked but has no specific gender anatomy. When physically fighting opponents, she has the capability to become invisible with a cloaking device to attack and defeat them.

The big corporation Megatech Body is a designer and manufacturer of Shells and has close ties to the government. Major is given the assignment of hunting down the notorious international hacker known as the Puppet Master. An assassination attempt by a mysterious recidivist foreign agent must be prevented. The Puppet Master cognitively manipulates small-time criminals. He is wanted by the law for espionage, terror, and stock market manipulation. There is a struggle going on within the police between Public Security Section 6 and Section 9 in New Port City. Major works for Section 9 and she handles a request from Chief Nakamura, the head of Section 6. Section 6 lures the Ghost of the Puppet Master into a specially created female Shell. The Ghost in the Shell wakes up, claims to be a sentient being, and oddly requests political asylum. At one point, Section 6 steals the captured body. Chief Daisuke Aramaki and his team of Section 9 look into the secretive Project 2501 and conclude that Section 6 created the Puppet Master for nefarious political purposes. They interrogate the Puppet Master who now appears visually as only head and shoulders, female breasts, upper torso, and truncated arms going down only to the elbows. Like the narrative that would find its way into the Hollywood version, Kusanagi's partner Batou saves her from death after her battle with a robotic spider-tank that leaves her nearly annihilated. Major's mind then gets connected to the

mind of the Puppet Master. The Puppet Master explains to Major that he was brought to life by Section 6. He wandered many cyberspace networks and became self-aware.

The Puppet Master makes a philosophical speech, contemplating existence. Humanity, he asserts, underestimated the implications and consequences of computerization. Yet the essence of life remains mortality, recovering each moment of the time that one has from one's future death. He wants to exist within a biological body that will die, to truly experience the human condition. Since he was owned by Section 6 within their network, this ambition of becoming human was not possible to realize, so he downloaded himself into a cybernetic body as the next best thing. He believes that Major is also "questioning her humanity." He knows this by having intermingled with her consciousness. In a previous scene, we observed that only when she was deep underwater while scuba diving, did she feel truly herself. In a speech in an elevator, she wonders if, since she is a full cyborg, her original self was not destroyed a long time ago. She speculates about her origin and if she only has an apparent simulated personality built over her cyber body and cyber mind. If her Ghost is also fake, then all human existence might be meaningless. If a Ghost or soul can be artificially built, as in the case of the Puppet Master, then humanism becomes definitively obsolete. The Puppet Master is an autonomous life form, born in the sea of information. He complains that he has feelings but is not complete. He does not wish to remain just a "copy," because copies are images that do not offer diversity or individuality. He wishes to merge his Ghost with that of Major. She agrees to the Merge. She will gain all his capabilities. Each will overcome their limits, become a part of the whole, face the bright light of the vast network, unify into a new structure. They merge their Ghosts. Major's Ghost becomes herself and what was the Puppet Master – now together.

Suddenly Section 6 attacks the building. They need to cover up the secretive Project 2501.

The Puppet Master's Shell is destroyed, but Batou saves Major's brain and its newly fusioned Ghost. The outcome is, at the end, that she gets a new cyborg child's body. She wakes up in Batou's home. The Puppet Master was not evil. She leaves the house. For the first time, her future is existentially open. Where her journey will now take her is unknown.