# HUMAN-MACHINE INTERFACE (1984)

By Frieder Nake

*“The development of communication itself is now the development of its means; in particular, its technical means.”*

**This article was first published in *Kursbuch* 75 (1984), pp. 109–118. Translated from German by Mari Matsutoya.**

Suggested citation:
Frieder Nake, Human-Machine Interface (1984). *Interface Critique Journal* 2 (2019), pp. 35–45.
DOI: 10.11588/ic.2019.2.66979

**In 1984, the computer arts pioneer Frieder Nake published "Schnittstelle Mensch-Maschine", a precise analysis of at the time young personal computer paradigms. The analysis does not only include the equally young GUI logics developed by researchers at Xerox PARC and first marketed with the Xerox Star workstation. It also reflects the implications for knowledge distribution and mental labour. Thus it makes sense that the article was published in the journal *Kursbuch* [vol. 75 (1984), pp. 109–118], which was established 1965 by Hans Magnus Enzensberger and Karl Markus Michel and soon became one of the leading periodicals for the German-speaking New Left. The specific volume is titled "Computerkultur" (computer culture) and includes, among other authors, Oswald Wiener, whose work is discussed by Nils Röller in this volume of *Interface Critique*.**

**We are indebted to the author for the permission to publish this valuable text from the childhood years of personal computer and GUI for the first time in English translation. We thank the ZKM | Center for Art and Media for the financial support enabling the translation of the German original.**

**Daniel Irrgang**

One of the leading lights of artificial intelligence, Edward Feigenbaum, states that the machines of the fifth generation[1] are not only possible, but that they are inevitable. In his opinion, computers will overtake the significance of the printed word. He sees a future where not just information, but knowledge of the highest quality will be accessible for everyone, everywhere, anytime. In an assumed allusion to Adam Smith, his colleague Pamela McCorduck says, "It is a future, where knowledge is the new wealth of nations".[2]

How Feigenbaum quite imagines the relations of knowledge, wealth and mediated labour, and where he would want to see a difference to the classic value-added production, remains open. One aspect seems sure – we are talking about the "machinisation"[3] of mental labour. The Japanese project can only be understood as an epochal attack on the intellectual labour that still lives. Perhaps the classic terms of work and working society implode over this attack and the vision of the omnipresent logical computer, perhaps not. In any case, one guesses that the critique of political economy is becoming more urgent for information processing. Has the processing of infor-

1 See Pamela McCorduck, Introduction to the fifth generation. *Communications of the ACM* (Association for Computing Machinery) 26/9 (1983), pp. 629–630. – North-Holland Publishing Co., Amsterdam, published the proceedings through a conference (1981) on the international announcement of the project.

2 McCorduck, Introduction to the fifth generation, p. 630.

3 I use this unusual term since the more familiar, "mechanisation", refers to a historic period only: that of the mechanical way of machinisation.

mation already so widely expanded for this critique to be adequately applied? Or should we still be content with fragments?

# Higher languages and lower ones

Computer science seems to be becoming the agricultural phase of mental labour. Just as increase in productivity lowers the overall value of the labour force, the value of mental labour force decreases with increasing productivity of programming work. At least as long as the use of computers remains an indispensable element of mental labour.

The use of computers and programs always means a transfer of mental work onto a machine. The living is transformed into dead mental work, the labour of the brain is turned into machinic operation. During a period of just forty years, this process has advanced with incredible force. In the beginning, the programmer would have to break down a formula, whose value was to be calculated, into a sequence of individual operations (addition, multiplication, etc.). That was not enough – he also had to code the operations (so instead of "+" perhaps to write "15"). He had to allocate space in the computer's memory to the values to which operations were to be applied. The numbers ("addresses") of these memory locations had to be substituted for the stored values.

For the simple equation "a+b", the command sequence would, perhaps, be as follows

| | | |
|---|---|---|
| *0100* | *10* | *("retrieve the number from memory location 100")* |
| *0102* | *15* | *("add the number in 102")* |

In the worst case, the programmer would have to calculate where on the magnetic drum he wanted to accommodate the commands in order to synchronise execution times of commands with the time needed by the drum's rotation. Imagine the relief of the programmer, when suddenly "a+b" could be written for the same expression. Also the capitalist, who applied the labour of our troubled programmer, will have felt relief but of a different nature.

This was made possible by "higher programming languages". They are the means for turning into machinic form such mental activities as mentioned here: breaking up formulas, allocating mstorage space, optimizing instruction sequences. Compilers translate programs of higher languages into lower ones, and are themselves programs. Nothing other than the clotted form of all the mental labour that goes into the said translation process.

The example of a mathematical formula does not suffice in illustrating the progress in productivity currently being highlighted. Let us, however, stay with a simple mathematical case. If x is to be determined to solve the quadratic equation: $a \cdot x^2 + b \cdot x + c = 0$, a series of "assignments" must be written at the traditional compiler language level, and be arranged into a proper sequence. In a language such as PASCAL, this could look like:

```
d := sqr(b) – 4*a*c;
if d≥0 then begin x1 := (–b + sqrt(d))/(2*a);
                  x2 := (–b – sqrt(d))/(2*a)
          end
else write ('no real solution possible')
```

It is plain to see that there is still a great deal of curious detail required to describe a very simple process in a high-level programming language. The "how" rather than the "what" is described, the language-related form rather than the problem-related content.

To change this, it should now be enough to specify only the relationships between the variables, and to demand the calculation of one of them.[4] For example:

$$a \cdot x^2 + b \cdot x + c = 0$$

*Compute x*

Machinisation of mental labour in general is often preceded by machinisation of programming work. Both the methodology of programming and ultimately its organization, as well as its means (such as languages), have changed profoundly in the last fifteen years. Endpoints are always formalized, generalized, and machinised activities.

In other words, the problem that the computer supposedly faces, calls for an adequate formulation of a linguistic level $L_I$. The computer is given a machine-language level $L_0$ (it lies deep as it requires a lot of detail). The machinisation of mental labour to some extent requires an intermediate level L (higher programming language). Translation from L to $L_0$ is machinised in a suitable compiler; $L_I$ to L must happen "in the head". Progress manifests itself in the raising of L. The above example shows this.

# Automated office work and Star

Let's turn to a current application of computer science, the automation of office work. The consequences of the evolution in the technical basis of office work is not the focus of our analysis. Let us rather take a look at current ideas for developers of such systems.

Generally speaking, the office worker deals with the preparation and design of texts and forms as well as the ordering, storing, copying, updating, evaluating and transmitting of received information. Such work can each be isolated and automated, but it also leads to isolated systems that may not be compatible. The real task of office automation is to integrate such isolated systems into a com-

4 The adept reader will not have missed that this is already possible in "functional" languages today. Further steps may be based on logical programming languages like PROLOG. It was defined in 1975 by the Frenchman Colmerauer. The Fifth Generation begins the with construction of PROLOG machines whose prototype are supposed to be completed in 1984.

prehensive one. By doing so, they hope to reduce the complexity of the system "interface" for the user.[5]

Three problem areas arise: distributed computer systems for office work that are only fully efficient as networks; simple yet complete "human-machine-interfaces"; "knowledge-based" systems.

The Star system of the Xerox Palo Alto Research Center is currently considered the most advanced office automation product on the market.[6] At well over 100,000DM, the price range seems to limit its distribution. Apple offers a system called Lisa with very similar features and at a significantly lower price of around 30,000DM, swiftly released by members of Apple's Xerox developer team.

Star is used by pointing to small iconic images on the screen, and occasionally typing text. Users do not have to memorize command sequences; they will always be informed of their options through what the system shows about itself. Should something need printing, e.g., an image of a printer would be pointed to; if calculation is required, a calculator image must be touched. If they tap on an image of a document and then on a paper basket, the document disappears.

Xerox invested about thirty "human-years" for the development of this "user-interface".[7] Its goal: "simple things should be simple; complex things should be possible". Or: it can be a pleasure to work with a system like Star; at the same time it lowers the necessary qualifications and makes jobs superfluous. At one bank in Hamburg, however, it is said that also higher ranking employees work with it.

# Where is the knowledge?

The real subsumption of mental work under capital has thus been initiated. In the automation of office work, it encounters the limits of its rule over purchased labour. As in timesof Taylorism, though at a higher level, automation managers and engineers must realize that they know nothing. Much to their regret, they realize that office work can only be described in an intuitive and informal way. The required formalisation for automation does not exist.

Even worse, American research indicates that the knowledge of office work is not found in the minds of individuals, but is spread amongst several.[8] Not only is office work itself more or less social, but knowledge about it as well. The work researched by Taylor was at least known to the individual worker, before it was elicited from them. In the office, however, capital seems to have to deal with the whole team.

For example, experienced office wor-

5 See depictions by C.A. Ellis and G. J. Nutt, Office Information Systems and Computer Science. *Computing Surveys* 12 (1980), pp. 27–60.

6 N. Meyrowitz and A. van Dam, Interactive Editing Systems. *Computing Surveys* 14 (1982), pp. 321–416, here p. 373.

7 Daniel E. Lipkie et al., Star Graphics. An Object-Oriented Implementation. *Computer Graphics* 16/3 (1982), pp. 115–124.

8 Ellis and Nutt, Office Information Systems and Computer Science, p. 53 f.

kers would introduce newcomers sometimes in a very indirect way to the specifics of their work. When negligence was observed, they would improve it in casual conversation in the form of jokes or side-notes. Equally, with employees of the same hierarchical level, informal communication about problems could be observed. In some offices there exists a loose but constant conversation between individual desks, which is not limited to private stories, but which also refers to the "cases" to be handled. A form of problem solving is cultivated, which essentially relies on immediate socialisation.

Automation of such work seeks to grasp existing complex social relationships in formal specifications and typifications and tears them apart. The mechanizers are terrified that it will be more difficult to maintain the efficiency of the office, which was intended to be increased. It is said that except in the field of text editing, no system has yet removed such informal structures.

# Human-Machine Interface

To speak of the "interface" between human and machine (computer), a system-theoretical approach is required. Systems are summaries of elements (components) between which relationships exist. The components can also be systems. Through abstraction, real systems become system-theoretical ones. This means in a given natural system, for instance, the components must first be identified as such. They are cut out. Surfaces of the cut emerge as "interfaces" to neighbouring components. The interface describes the relationship that should be maintained between the separate components, despite their disconnection. Technical systems are constructed. They can therefore be assembled from prefabricated components. Successful assembly requires the precise definition of the places, that are perhaps standardized, along which they can be assembled.

Systems in which humans and machines exist are hermaphrodites. Necessarily in these systems, humans are reduced to a few functions, a function-bundle. The automators, as one can imagine, don't really care. In fact, system analysis is an essential key to their approach. The straightforward talk of the human-machine interface proves that. Fortunately, there are others. Who would ever think of reading their car manual as the description of a human-car interface?

But let us not make it too easy for ourselves. As complex as a car may be in detail, its function is simple compared to a computer of around the same price. The car is a machine for converting fossil energy into motion, through which the load and the machine itself change their location. The operator must move with it as he monitors and controls the whole process, and he can abort and change it at any time. If we ignore operating errors and external influences, this machine is under complete control of its user.

The computer, on the other hand, can – by being appropriately programmed – be prepared to perform any practical, computable function (for us this means a mental activity that can be formalized). In the field of formalizing mental activities, the computer is the universal machine. It will convert any data you like according to a given program. Only the user who is also a programmer and knows the system, can think of total control. For anyone else, the machine will provide surprises.

The object of mental activity, as that of its corresponding programs, is information (not energy or matter). So then it becomes interesting at the "interface", if the program is interactive – that is, if there are open locations where the program interrupts itself, requires information from the user and only continues if and when these become available. Their openness gives the user the possibility of control. Or so it seems?

The design of this "interface" is a central topic of the ever wider use of computers. The machinisation will falter if the "interface" is not robust, clear, easy-to-understand and yet richly developed. The function of the "interface" is communicative: what humans and computers exchange through them is information. But the exchange turns out to be highly asymmetrical on closer inspection – what is information on the human side, is only data on the computer's side.

It is useful to remind ourselves of the concept of the "sign" as a three-point relation of means, signified, and interpreted[9]. The sign establishes arelation between a signifying means,a designated object, and an interpreted (subject matter (for an interested interpreter). Syntactics, semantics, and pragmatics examine the individual aspects of the sign. "Data" in this sense are signs as means, in computer-adequate form, encoded and stored. "Information" is gained from data only through interpretation, through assignment of objective as well as subjective meaning. Information is only bound to data, but data without information is uninteresting.

If, as a user of an interactive computer system, a person enters numbers and letters via a keyboard, the resulting data are of primary importance to him, and thus they become information. The moment these data penetrate the "human-machine interface", they lose this information to their new interpreter, the computer. They are reduced to their data core. The computer, under the direction of the program, uses them for storage entries or decisions. The latter "mean" nothing other than branching off in the program. The former "mean" nothing other than assignments of values to parameters. Only in this sense, data also gain meaning for the computer. This meaning is a different one, not least a narrower one, to the one that applies to the user. The fact that data within the computer acquire this meaning has been determined by the programmer in advance and not otherwise. The "interface" is therefore a place of lane change from wide to narrow. That is its communicative yet restrictive function.

9 In the German original text, the "interpreted" (as the result of an activity of interpretation) is wrongly called the "interpreter". Charles S. Peirce calls this the "interpretant".

# Communication between human and machine?

Each program is a static (namely, textual) description of a class of dynamic processes (its particularity is to process information at the reduced level of data). The single process is selected from its class by setting parameter values. It is executed when a computer interprets the program along with the values of the parameters. What the programmer has and has not included in the description of this class of dynamic processes defines the meanings that can be obtained in the context of the interactive program run.

The partners that come into contact through the communicative interface are much less frequently the computer and its user than the user and the programmer. Their means of communication is the computer and its program. This means gives communication the distorted appearance of "human-machine communication".

The process is simple enough and is more and more often seen as such. It must be all the more astonishing that even leading experts do not tire of packing it into anthropomorphizing forms. With some, one has the impression that they do it to get their hands on millions of research funding.

This immortal *Mensch-Maschine-Kommunikation* – human-machine communication – is sometimes referred to simply as MMK. It appears as "symbiosis" between human and machine[10] or even as "symbiotic tool"[11]. One wonders where the benefits might be that the tool hopes (to be used) for. The "self-explanatory tool" on the other hand, looks harmless, even downright technical.

The talk is of "convivial tools"[12], in a nice simplification of Illich's catchphrase that shaped society.[13] And what is tool on the one hand, is on the other, or simultaneously, partner; a partner who not only has an inner model of itself, but also builds one of humans, who are its partner.[14] The computer is asked to follow the principle, "Do what I mean, not what I say". Probably in the user's insight that he cannot clearly express anyway, quite unlike what Wittgenstein had imagined.[15]

I remain completely silent on the sorts of intelligence that break out of the interface, and on the imperceptible transfer of data and databanks to knowledge and

---

10 E. g. C. Berner, Die neue Symbiose: Mensch und Multiterminal. *Computer Magazin* 4 (1981), pp. 58–63.

11 Gerhard Fischer, Intelligente Benutzerschnittstellen, in: *Proceedings des GACM: Tutorials Intelligenztechnologie* (Stuttgart 1983). pp. 116–133.

12 Gerhard Fischer, Computer als konviviale Werkzeuge, in: *Proceedings der Jahrestagung der Gesellschaft für Informatik (München)* (Berlin, Heidelberg and New York 1981), pp. 409–416.

13 Ivan Illich, *Selbstbegrenzung* (Reinbek/Hamburg 1980).

14 On the critique of this frequently encountered position, see Ingbert Kupka, Susanne Maaß and Horst Oberquelle, Kommunikation – ein Grundbegriff für die Informatik. *Mitteilung* 91, Computer science department, University of Hamburg, August 1981.

15 Ludwig Wittgenstein, *Tractatus logico-philosophicus* (Frankfurt/Main 1963).

knowledge banks, perhaps in the hunt for research funding in looking to Japan.

I do not consider criticism of such usage idle. Conceptualization is preceded by absence of concept, which may well come along with powerful words and yet does not get any better. If a science like computer science contributes to such drastic changes of work and life, as it turns out, and if it tolerates or even promotes such a casual approach to its conceptualization, it serves – intentionally or otherwise – the veiling of real circumstances and changes. Let's then take another look at "human-machine communication".

It seems relatively easy to identify the root of the false consciousness expressed in the crooked conceptualization. Communication seems to be a very early achievement of humanity. Mumford goes so far as to set its significance for the process of becoming human higher than that of tool-making and use.[16] Dialogue is the elementary form of communication. That does not mean that it also historically came first. However, it has all the features necessary for communication. It is originally characterized by a unity of place, time, and participants. This is already conditioned by the first means of communication, the voice.

The development of communication itself has become the development of its means; in particular, its technical means. They break up the original unity in multiple ways, and muscle in between the participants. Written text removes the unity of time and allows a limited form of communication even with past generations. The phone removes the unity of the place and potentially extends communicative options to all living beings. In many cases, the place and time of the communication are extended.

The computer as a currently last stage of development of technical means of communication also eliminates the unity of the participants: they are distributed, instead of their own thinking, some *thing* is "thinking", it is "communicated". This is possible due to the specific nature of the computer, the processing of information in the form of data. The means of communication here is not limited to the transmission of information by a (largely) constant information carrier, but it is able to change this carrier, the data. This leads, with appropriately advanced programming, to the impression that the means of communication have become independent, that they themselves have become the partner of communication.

The truth of the process is that a communication partner ("user") usually enters into a multiply fractured dialogue with a whole group of partners. They do not know anything about the concrete communication. They have pre-formulated questions in the form of programs and systems of programs, and answers stored as data or algorithmically determined. The user calls on, so to speak, only one or the other answer from a possible variety of dialogues, which may be infinite. The dissolution of the unity of the participants is also reflected in the fact that the programmers have plan-

16 Lewis Mumford, *Mythos der Maschine* (Frankfurt/Main 1977).

ned these dialogues only as parts, not as whole progressions.

"Human-machine communication" turns out to be a helpless formula for a deeply social process.

Michael Paetau rightly points out that "the involvement of the computer in interpersonal communication is first of all, a change in the form of communication caused by a new medium"[17]. The forms of communication are subject to historical changes that are shaped by the level of development of productive power. If you look at it this way, one gains reasonable access to the process of human-machine communication which may then even be named as such.

Or do such people see far ahead into a bright future, in which work essentially means communication, and wealth comes from knowledge – those who seem to conceptually shape concepts? Do they look to a future where work is only done by machines, and where humans reproduce by training a machine to produce and, for that purpose, communicate quite naturally and with ease?

Is the machine, perhaps, repudiating its way of being, the appearance of fixed, constant capital, the mode of existence in which it originated and in which we experience it? Does the machine as an information machine emancipate itself from its capital form and lead the astonished worker into a future he could never have created?

Crazy thoughts, if one thinks of the simultaneous debate on the new analysis of the relationship between productive forces and the relations of production, on the question of whether the productive forces are in part indelibly imbued with capital. There is no real reason to follow such thoughts. After all, software (programs) is by nature (description of information-processing processes) nothing but "work organization cast into technical functional mechanisms"[18]. In this way, the inherent claim to domination can be traced more easily than in other machines.

André Gorz points out that "automation in itself is socially ambivalent", and that "microelectronics is an 'open' technology"[19]. The socialization of labour, in the case of machinisation of mental labour, seems to be near. Also near seems to be its contradictory stance to capitalist relations of property. The "human-machine communication" removes the producer even further from his product, which exists for him only as a description. At the same time, this abstraction opens up hitherto unattainable work areas for him. Is the socialization of the means of production not actually overdue?

17 Michael Paetau, Soziologische Dimensionen computergestützter Bürokommunikation, academic paper from *GMD* Nr. 18 (Bonn, March 1983), p. 15.

18 Ibid, p. 43.

19 André Gorz, *Wege ins Paradies* (Berlin 1983), p. 49

# References

**Berner, C.**, Die neue Symbiose: Mensch und Multiterminal. *Computer Magazin* 4 (1981), pp. 58–63.

**Ellis, C.A. and G. J. Nutt**, Office Information Systems and Computer Science. *Computing Surveys* 12 (1980), pp. 27–60.

**Fischer, Gerhard**, Computer als konviviale Werkzeuge, in: *Proceedings der Jahrestagung der Gesellschaft für Informatik (München)* (Berlin, Heidelberg and New York: Springer, 1981), pp. 409–416.

**Fischer, Gerhard**, Intelligente Benutzerschnittstellen, in: *Proceedings des GACM: Tutorials Intelligenztechnologie* (Stuttgart: Teubner), pp. 116–133.

**Gorz, André**, *Wege ins Paradies* (Berlin: Rotbuch, 1983),

**Illich, Ivan**, *Selbstbegrenzung* (Reinbek/Hamburg: Rowohlt 1980).

**Kupka, Ingbert, Susanne Maaß and Horst Oberquelle**, Kommunikation – ein Grundbegriff für die Informatik. *Mitteilung* 91, Computer science department, University of Hamburg, August 1981.

**Lipkie, Daniel E. et al.**, Star Graphics. An Object-Oriented Implementation. *Computer Graphics* 16/3 (1982), pp. 115–124.

**McCorduck, Pamela**, Introduction to the fifth generation. *Communications of the ACM* 26/9 (1983), pp. 629–630.

**Meyrowitz, N. and A. van Dam**, Interactive Editing Systems. *Computing Surveys* 14 (1982), pp. 321–416.

**Mumford, Lewis**, *Mythos der Maschine* (Frankfurt/Main: Fischer, 1977).

**Paetau, Michael**, Soziologische Dimensionen computergestützter Bürokommunikation, academic paper from *GMD* Nr. 18 (Bonn: GMD, March 1983).

**Wittgenstein, Ludwig**, *Tractatus logico-philosophicus* (Frankfurt/Main: Suhrkamp, 1963).