lence. We may once have believed that we had control over life, but this control is slipping, and we must steel ourselves to endure the countless disasters we may have to face. The Stoic philosopher Seneca, who lived in a time of similar uncertainty, argued that we should decrease our vulnerability to shock by meditating daily on everything that might go wrong. "Mortal have you been born; to mortals you have given birth. So you must reckon on everything, expect everything." Drama can serve as a premeditation in Seneca's sense. The dramas of daily life in ego-culture, ego-centricity and socio-centricity, the precariousness of the future of Man and the other species, whether linked to natural disasters or created by human intervention, and the new threat of global death, are so many sources for a therapy based on the dramatic arts.

The capacity for endurance is not the same as fatalism, and it does not imply that we are unable to effect change. Indeed, if we are to react effectively, we must keep our emotional responses alive. The bringing out of emotion associated with drama may be our best resource, training us in empathy and inspiring the kind of solidarity we will need to face the challenges before us. We need to remember that humanity is one tribe, floating through space on a planet that has come to resemble the Raft of the Medusa.

3. Dialogue with Machines
Can Computers Be Interactive?

Peter Krieg

If by "dialogue" we mean dialogic reasoning as in "Socratic dialogue", today's computers are not capable of it: their interactivity is limited to strictly obeying predefined commands and answering simple queries within a single logic domain. This mode of communication is not even at the linguistic level of the stone ages. The issue here is whether computers based on closed logic are principally able to develop a genuine ability for dialogue. The author describes a new "polylogic" approach which holds potential for innovative cognitive computing and dialogic applications in communication and the arts.

"Man, when he is not an automaton, is not functioning properly."

– Antonio Porchia

We have gotten into the habit of describing our interaction with computers "dialogues with machines", thus equating the interaction between man and computer with language based dialogue between intelligent knowledge systems capable of learning. Yet very crucial differences exist here.

Language based communication probably emerged first as a means of coordinating prehistoric hunting groups via signals, sounds and signs. They got by with commands and shouts. At least since Socrates, dialogue (combining the Greek words for "two" and "knowledge") refers to a complex process of linguistic human communication, where arguments from different speakers are integrated and new conclusions synthesized, from which one can logically deduct analytic conclusions, which again can be reintroduced into the cycle as new arguments in a higher order. This process has also been described as "collegial reasoning" (Van Hooft, 1999) and it combines synthetic and analytic operations as well as communication. In natural languages this process can be continued to the point of abstraction where we can use language to speak about language. Moreover, we integrate all language contexts in this process, regardless whether they are directly connected to the theme of the dialogue and thus seem relevant to it or not. We can connect contexts freely via comparisons, metaphors and analogies. This very specific human capacity of linguistic abstraction and concretization, of free association and stratification, of synthetic and analytic operations in its entirety – and only in its entirety- constitutes human thinking and has probably enabled the development of self-consciousness and of what we describe today as human identity.

Neither the ability to make global associative, cross-contextual connections nor of conducting a dialogue in arbitrary abstractions and dimensions are generally available in computers today. Their underlying architecture is logically closed, i. e. connection is restricted to one logic domain as frame of reference – e.g. documents in one format or type sharing the same lifetime. Even if we could use a very large working memory able to hold all data of the system, we would still be restricted to a one dimensional Turing machine with one frame of reference only. In addition a data based "associative memory" would be prohibitively expensive. Data would pile up exponentially according the increase of details available for combinatory operations.

While a slightly different method is used on the Internet, where all data are accessed through independent roots (URLs) as containers of their respective hierarchical pages holding the actual data, it does not qualitatively change the result: data are invisibly encapsulated in hierarchical structures and cannot be compared and interconnected. Search engines try to overcome this problem by downloading and indexing parts of the internet on their own servers. Like index schemes in general, this does not cure the disease, but at least eases the pain somewhat … Internet search engines allow only a limited number of words and arguments as search query (usually no more than 10 words). The results cannot be returned as new queries in a next higher order of search. It is up to the user to evaluate, redefine and especially reduce. Internet browsers and search engines therefore still serve as trivial navigation and orientation systems, like electronic library catalogues, connecting the user to containers of redundantly recorded and hierarchically structured data of a traditional library. Search engines usually can only answer questions in the form of "where (in which container) do I possibly find an answer?", but cannot give a direct answer to a concrete question.

With current computer architectures, dialogic processes are furthermore restricted by factors like the number of possible dimensions as degrees of "computational complexity". High end computers today are supposed to cope with databases having up to 14 dimensions, yet at high computational and financial costs.

A knowledge system is far from a simple library that provides access to latent knowledge in the form of arguments residing in books. A knowledge system must be capable of analyzing all represented data and synthesize new knowledge from it. Such new knowledge, for example, could consist of predictions or arguments derived from comparing existing arguments across contexts (including non-relevant ones). Such a system would be capable of learning in the sense of being able to create new hypotheses in new contexts on the basis of its former successful hypotheses in other contexts. Computer systems capable of learning in this way do not yet exist.

Computers today are still "trivial" machines – if we disregard the inherent non-triviality of all physical systems in an entropic universe. They are constructed as linear chains of dependencies of their parts, as mechanisms that follow hierarchical logic operations in order to reliably compute a specific output from a specific input in predictable and predetermined steps. Even their ability to generate random numbers is, strictly speaking, simulated and based on linear programs. Only the user unacquainted with the program or unable to see through it considers it random. Hackers prove time and again that computer programs including encryption programs are trivial and deterministic – and therefore can principally be cracked with enough computer power. Since software as well as data are hierarchically structured as "if–then" chains of dependencies, they are structurally deterministic: The beginning ("root") of a hierarchical structure encapsulates the complete knowledge about its subordinate layers.

Logic as a hierarchical system of deductive inference has only one perspective as way of looking at the things it contains (which are also the only things it can look at). If a thing belonging to one logic domain at the same time also belongs to another one, the two logic domains intersect in the thing and create complexity in the form of ambivalence: the very same thing can be logically "true" in one domain and "false" in another. Words in natural languages, e.g. are always ambivalent for good reasons. A "plant" belongs to the domain of biological systems, but also to the domain of buildings. Depending on the context, a sentence referring to a plant can be false in the domain of biological systems, but true in the domain of buildings.

Systems capable of learning must be able to map arbitrary including unknown signal patterns, to generate arbitrary logic domains as hypotheses and to compare all patterns and contexts irrespective of format, standard or type. Comparing different types of things requires a polylogic approach, because in a mono-logic system we can only compare one type. Adaptive, learning systems therefore require a polylogic system of mapping signals to a reference structure and a separate input structure. One looks to the inner, already represented pattern of the system as normative reference, while the other looks at the outside, the input patterns, as a cognitive reference.

We can describe learning systems as continually blending, integrating and correlating patterns from external and internal events. In order to create new knowledge and to change perspectives – which is what learning is about, they must map representations of these patterns into a transparent internal map that allows to "remember" patterns as past experiences as well as compare and reproduce patterns. This complex map must allow to specify logic domains as hierarchies, but also to fluently transcend these domains. Therefore its architecture cannot be just hierarchical, it must also be heterarchical. Only a polylogic architecture can support both structures at the same time and switch at any point from one to the other. Knowledge systems, in other words, are not conditioned by a single perspective. They are not *complicated* logic mechanisms, but *complex* polylogic systems.

What we observe as a process of learning in a biological system is a behavior that continuously generates a dynamic balance between its structural limitations and environmental perturbations. It is this cyclic correlation process that produces "meaning" because it transforms signals into information as "a difference that makes a difference" to the organism (Gregory Bateson). Meaning does not exist independently but only as relation to success or failure, life or death of a cognitive system: if the system loses its balance and becomes chaotic its organization disintegrates and it ceases to exist as a system. In logic terms, learning systems continuously differentiate and integrate the many logic domains they generate as part of their interaction with the environment to produce a dynamic balance which the observer may interpret as predictability. Some call this adaptive process "self-organization"; the Chilean

neurobiologist Humberto Maturana coined the term "autopoiesis". Such an autopoietic, polylogic machine is probably also what Antonio Porchia had in mind when he called man an "automaton".

Humans can simulate such autopoietic processes in language. Natural languages can describe, model and communicate cognitive processes: in language we also make cross-contextual comparisons in order to create new patterns based on analogical "intuitive" inferences, from which we then can analytically infer. In dialogue we interact with an unknown environment represented by another human being that we principally cannot analyze (because it also is a non-trivial system) but treat as predictable because it is structurally similar to us and thus creates similar states of stability as attractors. Socrates considered dialogue the most important method to build knowledge – not because knowledge could be transferred via language, but because language simulates cognitive adaptive behavior.

The idea of language as a means of "transporting" instructive information or knowledge is a variant of the idea of "true" representation ("Abbildungstheorie") of the environment in the mind. Although still represented even by some scientists, it can neither explain the evolutionary rise of cognition (when and how could the knowledge as basis for the interpretation of a such representations have developed in evolution?) nor can it solve the problem of infinite regress by locating the ultimate instance of interpretation (the interpreter of the interpreter) and the source of its knowledge. But this also means that language is not a container of knowledge or meaning. It is just a complex human behavior among many others except that it uses linguistic conventions to symbolize arguments.

It can also be argued that for the same reason cognitive systems do not record data, but generate data as an emergent result of their operations. When a machine records sound or images or text it does so by representing every occurrence of the same pattern again. This could be called an "direct representation" and is common to both analog data recorders (like tape recorders or film cameras) and digital recording devices like computers, digital cameras or digital sound recorders. Such a redundant data recording scheme does not, however, automatically form a complex, polylogic and interconnected pattern structure capable of self-evaluation but requires predefined orders allowing to store and retrieve the data later. In biological systems no such orderly storage structures exist, nor do natural environments exhibit redundancies in the form of recurring identical events or things. A biological system would therefore have to record everything it perceives and could not even compare it to other things or events due to the lack of a reference. No biological cognitive system could even hold such an amount of data. Therefore we may ask whether a cognitive computer would also require a similar data representation scheme.

The issues of data representation, logic closure and complex polylogic mapping are central issues of artificial intelligence (AI): Since 50 years now AI has failed to construct "intelligent agents", "complex adaptive systems", autonomous robots or in general cognitive computers. With the exception of Neural Networks AI is based on the assumption that the complexity of the environment can be reduced to logically consistent mathematical models which can be implemented by traditional software architectures to solve complex problems.

If we define complexity as a description of events with intersecting logic domains, and cognition as the ability to integrate different logic domains by mapping their representations onto a complex polylogic map, then dialogue requires such complex mapping on both ends. In order to conduct a dialogue with a machine, the machine must also be able to adapt to the linguistic complexity of the human partner by mapping arbitrary linguistic domains into a globally interconnected transparent space that it can self-evaluate rapidly. These evaluations must also be able to extend to all patterns represented in the system. As a result of these evaluations, the system eventually even should be able to transcend any pre-knowledge it might have (e.g. as pre-programmed algorithm) and to generate new algorithms and patterns. This would constitute an "artificial cognition" system with creative abilities to synthesize new perspectives, hypotheses and axioms and analyze them with its own algorithms. Again, this would only create meaning for the machine, not necessarily for the human user. But it would closely simulate a human process of generating language based knowledge and meaning and could offer the human dialogue partner a result that he/she can more easily convert to his/her own individual meaning. In this sense, the machine would behave like a human dialogue partner.

Computers are not yet capable of doing this. From "Eliza," Josef Weizenbaum's famous mock dialogue program, to "Deep Blue," IBM's chess mainframe, which was able to defeat the reigning world champion, all we have been dealing with until now are just high-speed mechanical-logical computers and programs of various degrees of complexity conducting largely linear, non-adaptive operations. These operations may appear to the observer as thoroughly "intelligent", but they actually merely compute and select, on the basis of preprogrammed and predefined models and lists, the appropriate actions for the given situation. In this way, they operate strictly within the limits of deterministic systems. Chess is especially well suited for this, since it is based on a small set of rules and thus operates also deterministically within a finite number of possible states. Because this number of possible states is so big, however, the computer can take advantage of its processing speed in scanning suitable options, especially if it also implements smart rules, successful proven game strategies and uses powerful hardware. There is little doubt that within a few computer generations, no Grand Master will have even a ghost of a chance against a modern chess computer.

If we still consider computers as being "dumb," this is chiefly because they have not yet become capable of genuine thinking and dialogue and thus can only operate analytically within predefined contexts. Thinking is always analytical *and* intuitive. Since Kant we also call these operations "synthetic" and "analytic". Synthetic operations are based on intuitive analogies between different patterns and aim at finding or constructing fitting logic domains as hypotheses for interpreting new patterns. Analytic operations then use rational logic to deduct conclusion from these perspectives. Only together they constitute a thinking cognitive system.

For us thinking is such a matter of fact that we are not usually aware of its complexity. Actually, we were told for the last 2500 years that correct, rational thinking was reserved for logic, analytical operations. Intuitive, synthetic thinking was considered primitive and irrational. Since computers can do these mechanical rational operations much faster and more reliable than us (our main strength lies in intuitive pattern recognition and discovery and synthetic thinking) we could easily be brainwashed in believing that computers are the superior thinkers with at least equal intelligence. So when we fail to communicate with a computer we often consider this to be our own fault and not the shortcomings of an inflexible computer logic. A machine like today's computer that only operates analytically within one given context can hardly qualify as a thinking machine at all. Even its interactivity only simulates the most primitive forms of dialogue by taking advantage of either its computing speed or the user's lack of knowledge with respect to its program routines.

In recent years computer games and virtual reality systems have brought some interesting new perspectives to this debate. What makes playing computer games quite exciting pastimes is primarily their direct, fast and interactive response to the human game partner that comes much closer to dialogue than language based computing. Although here too the human is the only adaptive, cognitive and intelligent player in the game, the flexibility and speed in which the machine can change its behavior and generate the corresponding images and sounds make the illusion of dialogue much more believable. The technical base for this is the fact that computer games have moved away from the traditional data representation scheme and generate their image data dynamically. They do not hold image data like a digital movie or slide show, but mainly virtual data in the form of algorithmic models that are animated and texture mapped interactively and dynamically on demand. While game software architecture is still based on traditional closed logic, the virtual data scheme allows much faster data generation than accessing files of rendered images on a hard disk or CD-ROM. Furthermore, since interactive games can generate practically unlimited variations of images and movie-like sequences, storing each possible frame would physically not be feasible. While a 2 hour digital DVD movie, even with strong compression, requires several Gigabytes, a game in similar resolution can be played for many hours from just a CD-ROM with a capacity of no more than 700 Megabytes.

The progress computer games have made in just a decade can be appreciated by comparing early interactive media trials. The author was involved in this history as founder of the "interActiva" festival for interactive media (1990–1996 in Cologne and Babelsberg) and as director of an interactive cinema experiment in 1990 that attempted to link gaming experiences to traditional linear narrative forms of the cinema, combining the collective viewing experience of a cinema audience with individual options to interactively control the story.

Here we will largely ignore the question as to what extent the linearity of storytelling might not actually represent an essential cultural and psychological technique to produce individual and social stability of cognitive and normative expectations required in a dynamic social environment. Instead we focus on the question of interactivity and dialog in man-machine communication. I will make special reference to my own experiments in Germany with one of the first interactive cinema systems which have also been confirmed by later, larger-scaled and more complex experiments in the United States.

The author's Interactive Cinema System (E.L.V.I.S. for Electronic Video Interactive System) gave the participating audience the option to collectively select "footnote" sequences at several points of a linear documentary movie presented from an analogue video disk. The availability of a footnote was signaled by a 10 second indicator bar appearing on the projection screen. Everyone in the audience had a simple computer game control pad with one active button only. When a majority of the audience pressed the button while the indicator was on screen, the footnote was selected and the film seamlessly switched to a second disk and continued with the footnote sequence. At the end of the footnote the system switched back to the main film. The audience was not able to distinguish between main film and footnote sequence.

The result was quite sobering for the audience, the filmmaker and the programmer: while the audience at first was highly motivated and excited about the experiment, the charm of novelty faded quickly. Almost without exception, it left the viewer at the end of the film with the frustrating impression of having missed something. Usually cinema pleasantly dissolves the everyday life frustration that the most interesting things always happen elsewhere. At least in the cinema everything that is of importance for a story happens before my very eyes on the big screen. Context and suggestive interpretations of the events are structured as a story with a beginning, a middle and an end. Interactivity departs from this principle and throws the

Fig. 3.1 Audience at an interactive cinema demonstration by Peter Krieg and Martin Frech, 1990

Fig. 3.2 Film poster display at Traumstern Cinema in Lich, Germany, one of the venues on the 25-city tour through Germany and Switzerland with a one-stop foray into the U.S. for the international public television broadcast conference INPUT in Baltimore

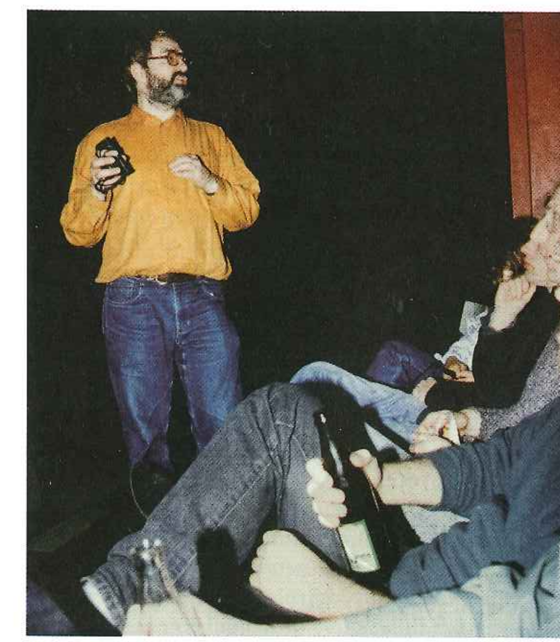

Fig. 3.3 The author briefing the audience

Fig. 3.4 Technical set up of the mobile interactive cinema: two analogue video disk players, 386-MS DOS PC, router, video projector and sound system

viewer back into precisely the chaotic continuum of everyday life that the audience was seeking to escape from. The principle of interactive cinema may appear to be superficially interactive, since the activities of the participants trigger new activities on the audio-visual level, but no real dialogue takes place. Instead, trivial choices are made from a tightly programmed catalogue of options with the help of various selection procedures (usually majority votes), where the number of possible film variants may be enormously large (in the above example, there were only 16 branchings in the film, but these provided 64,000 possible film versions).

Compared with today's computer games, this is a ridiculously small range of variation, but the main effect was related to the voting mechanism: since each member of the audience at some point ended up in the minority and thus did not see all of the footnote sequences that he or she had voted for, almost everyone came away with the feeling that they had been cheated out of perhaps the best sequences through the voting behavior of the others. Interactive cinema proved a "frustration machine": the longer the film continued, the greater the number of frustrated viewers it generated. So much for the entertainment value of continuous voting.

Consequently, these early attempts of interactive cinema quickly became extinct and were replaced by internet and LAN based network games where each individual player is in control of an avatar representing the player as a fully active protagonist in the game.

Fig. 3.5 The author handing out control pads

Fig. 3.6 Participants in action (one should note the way most viewers are leaning forward, which reveals a psychophysical state different from that found in a normal film)

The rapidly increasing graphic capabilities of game computers are directly related to their very fundamental break with traditional data representations schemes: generating movie images dynamically instead of playing back stored images from disks allows them to react in real time to the interventions of the players. Current graphic chips and engines already generate video at high definition resolutions, thus rapidly closing the gap between the image quality of video or computer generated films and dynamically generated interactive games. As interactivity gets more and more fluid, the expectations and demands of users for more cognitive dialogues will increase.

While computer games are structurally still using non-adaptive, closed logic architectures, their virtual data paradigm is a giant step towards a cognitive platform capable to generate dialogues, because it principally solves the important issue of storage capacity and access speed. Traditional redundant data representation creates mountains of poorly connected data layers that usually have to be managed and accessed by navigating endlessly branching, hierarchical tree structures. But once all data and even data structures are only virtually represented and generated by fast algorithms like is done in a game engine, no more shoveling of data as "real estate" will be necessary. Could the combination of a polylogic architecture with a virtual data scheme be the royal road to cognitive computing and true man-machine dialogue?

The inventor Erez Elul has recently developed a new software architecture ("Pile System") that generically and automatically represents all data not as redundant collections of symbols but exclusively as connections which can be grounded in values representing arbitrary codes (like ASCII). Any pattern this systems sees as input sequence and sub-sequence is related either to a connection already representing the same sequence seen before, or, if it is a new pattern, represented by new connections. In this way, all sequences – e.g. words – are connected to each other as well as to their respective neighbors in their input sequence. Although a word or sequence (string) can appear many times within the entire text documents of a system, it needs to be represented only once as a series of connections. For any recurrence, one new connection pointing to the original representation is required. Since every symbol is represented by a separate logic domain as root, the system has many independent roots and each representation (connection) is connected to at least two roots. The term "polylogic" refers to this requirement that every system object (= connection) belongs to more than one root. This is quite different from traditional logic structures where an object has only one parent under one root. As a result, the system holds only complex connections as its objects. This has several quite amazing and far reaching effects:

- the entire representation space of the system is fully interconnected enabling a self-reflexive global associative memory.
- no traditional data are recorded. The system only holds connections as "virtual data". "Real data" are dynamically generated like a computer game engine generates its image data.
- the system treats data, data structures and codes in the same way, i.e. data structures are virtual as well and are associated to data dynamically. A database thus is a dynamic simulation like a specific computer game situation.
- since data and structures only exist virtually, the system does not restrict data sizes, dimensions or degrees of complexity. To query the system, e.g. any query size can be used without performance cost. In contrast today's search engines usually limit their queries to a maximum of 10 words.

If a software architecture has no size or dimensional restrictions, it is considered scalable. If all its representations are visible and interconnected, it is considered self-reflexive. If it allows complex representations of arbitrary logic domains it is considered complex. If it is complex, self-reflexive and scalable, it should allow cognitive properties like complex adaptive systems, learning systems and systems able to conduct Socratic dialogues.

The fundamental restriction of today's computers to one single logic domain at a time forces the user into a rigid logic that he or she has to adapt in order to be able to interact with the machine. Instead of the machine adapting to the individual user, it only interfaces to an imbedded model of a statistical "common user". As long as we individually behave like this common user, the illusion of a dialogue with the machine is sustained. But the moment we employ our personal idiosyncrasies, the dialogue abruptly ends. "Personal computing" can quickly turn into a very impersonal experience.

There have been many attempts to soften this uncomfortable relation, ranging from "user friendly" tools to "fuzzy logic". Yet none of these departs from the traditional logic closure: User friendliness is euphemism based on the employment of adjustable graphic user interfaces and visual gimmicks like buttons, colors and animations. Yet these largely decorative and admittedly often helpful band aids do not really change the underlying architecture. They do, however, drastically increase computing costs in the form of hardware requirement, memory use and software complexity/size. As a result, software has rapidly grown bloated and brittled, without adding much functionality except better graphic capabilities.

Fuzzy logic has been hailed as another solution, but it also is not a departure from closed logic: it only provides a range of true/false possibilities within a single logic domain. While offering more system tolerance, it does not support adaptivity or dialogue.

The original promise of computer technology was a new type of cognitive machine able to serve as an extended "electronic brain", much like classical Archimedean machines extended human muscles, hands or legs. Yet despite all announcements to the contrary, computers are still no more than Archimedean machines. If we agree that thinking is a cognitive process combining synthetic and analytical operations, then computers still do not think at all. They possess neither intelligence nor consciousness, and all popular visions of computers taking over the world and "not needing us any more" are not only exaggerations but fundamental misconceptions.

The dominating approach of computer science and industry is still to simulate cognitive functions using non-cognitive techniques. One of these attempts is currently implemented on the internet in order to better connect the semantic context to a relevant, specific query. The method used is "semantic tagging". This technology (XML) is based on traditional indexes that relate a word, sentence, paragraph or document to a larger semantic context. These standardized contexts are published as manuals. The problem here is that the tag again is only one-dimensional (connecting the tagged item to just one context) and good tagging requires "structured writing" from the author of the document. To do this properly, the author must consult manuals with thousands of pages. Automatic indexing is being used also, but it is only as correct as current automatic, mostly AI based tools can be.

While hardware technology – especially in microprocessors – has constantly succeeded to implement higher performance, integration and complexity following "Moore's Law" (actually a trend, not a law), software lags behind as long as it is stuck in the paradigms of the past. While every compact car is said to have a thousand times the compute power of the lunar landing modules of the Apollo program, this immense hardware progress has not been translated into equally increasing software functionality. Quite the opposite: software today is considered the main risk factor of practically all technical systems. Writing this text on my notebook required less than 100 MHz processor speed and hardly any memory a few years ago. Nowadays one can hardly buy a desktop machine with less than 1 GHz processing speed, and the memory required to just run the operating system and basic application software is more than the capacity of the average computer's hard disk some 5–6 years ago. Nevertheless we still basically type, erase, cut and paste.

The promise of the computer to simulate and execute cognitive operations, to automate simple and recursive tasks, to evaluate all its data representations and to create new information as a basis for better human decision making – all of these promises have not been fulfilled. Instead of the "electronic brain" we got an electro-mechanical Skinner box. Computers can claim – like the famous laboratory rat – that they have successfully trained their masters to throw food into the cage whenever the rat pushes the red button. We have been reduced to little more than computer peripherals.

The era of the interactive computer capable of dialogue is still to come. Today's rudimentary applications such as computer games, interactive computer art or video-on-demand, are rapidly reaching the peak of their technical potential limited by traditional architectures. The qualitative leap in an artistic, psychological, philosophical, and technical sense will be from today's control medium to a true dialogic medium which opens completely new possibilities for interactivity.

These systems will allow, i. e. to invite Shakespeare to dinner: based on his writings, this machine will be able to combine an interactive three-dimensional animation of the playwright, to engage in a synthetic and synchronized speech conversation with a human, and to generate sensible arguments and statements in response to arbitrary questions or arguments of the human dialogue partner. The conversations of a "Virtual Shakespeare" will not necessarily be restricted to literal quotes from his works, but will be newly synthesized arguments based on the linguistic and semantic patterns of his writings without necessarily containing any literal quotes. Imagine the schoolbooks and university textbooks of the future as virtual teachers – digital reincarnations of famous scientists, historians, writers, or teachers. Computer games would similarly provide dialogues and interactions between cognitive and adaptive systems on both sides creating new challenges and opening new dimensions of entertainment.

Another vision is to integrate the entire Internet data in a cognitive architecture as global knowledge system that operates with semantic and linguistic patterns: it could deliver direct answers to direct questions in a Socratic dialogue without the current detour of having to manually scan hundred or even thousands of pages for possible answers that might be hidden somewhere. Each such dialogue would create new patterns enhancing the knowledge available in the system.

A cognitive "polylogic" computer paradigm can be a powerful computing platform beyond the mechanistic and closed logic of today's machines. While programmers and computer scientists – who were trained and conditioned to think exclusively in logic terms – might have problems descending down from their tree structures, artists as well as philosophers and most users have insisted on creative, adaptive and non-trivial systems. They will quickly seize the opportunity of having a true personal machine supporting and extending their individual creativity. They will help us to immerse ourselves in adaptive cyber worlds that integrate machine-generated art and knowledge with fictional elements and plot lines as well as with our personal biographies, fantasies, and artistic expressions.

But even a cognitive computer cannot duplicate the specifical human form of creativity and intelligence. Meaning also will still be exclusively a product of the listener, not the speaker – be it human or machine. But the cognitive computer will be a true extension of our own cognitive operations and thus will represent a part of our consciousness.

References

3.1 Elul, Erez: A System Of Naming (Berlin 2002), www.pilesys.com
3.2 Krieg, Peter: Suspicious Minds or The Order of Chaos, linear Super 16 mm film and non-linear interactive video disk (Cologne 1990)
3.3 Krieg, Peter: Beyond Paranoid Computing (Berlin 2003), www.pilesys.com
3.4 Porchia, Antonio: Voces (Cologne 1978)
3.5 Van Hooft, Stan: Socratic Dialogue as Collegial Reasoning, Practical Philosophy 2.2, July 1999

Part I | 3

Part I | 3