Some Implications of Comparing Brain and Computer Processing[1]

Brian Whitworth
Massey University, Auckland, New Zealand

Abstract

Like a computer, the human brain inputs, processes, stores and outputs information. Yet the brain evolved along different design principles from those of the Von Neumann architecture that lies behind most computers in operation today. A comparison of human and computer information processing styles suggests basic differences in: 1. Control (Central vs. Distributed), 2. Input (Sequential vs. Parallel), 3. Output (Exclusive vs. Overlaid), 4. Storage (by Address vs. by Content), 5. Initiation (Input vs. Process driven) and 6. Self Processing (Low vs. High). The conclusion is that the brain is a different type of information processor, not an inferior one. This suggests replacing technological utopianism with socio-technical progress, where computers plus people form more powerful systems than either alone. For this to occur, the computer must change its role from clever actor to simple assistant.

1. Introduction

Over thirty years ago, TV shows from The Jetsons to Star Trek suggested that by the millennium's end computers would read, talk, recognize, walk, converse, think and maybe even feel. Since people do these things easily, how hard could it be? Yet today we generally still don't talk to our computers, cars or houses, and they still don't talk to us. The Roomba, a successful household robot, is a functional flat round machine that neither talks to nor knows its owner. Its "smart" programming mainly tries not to get "stuck", which it still frequently does, either by jamming itself under a bed or tangling itself in carpet tassels.

Computers do easily calculation tasks that people find hard, but equally the opposite applies, e.g. Figure 1 shows a Letraset page which any small child would see as letter As, but most computers have difficulty with this. Other tasks easy for people but hard for computers include language recognition, problem solving, social interaction and spatial coordination. While people recognize familiar faces under most conditions, computers struggle to recognize known terrorist faces at airport check-ins because variations like lighting, facial angle or expression, or accessories like glasses or hat, upset the computer's fragile logic. Advanced computers struggle with skills most five-year olds have already mastered, like talking, reading, conversing and running: "*As yet, no computer-controlled robot could begin to compete with even a young child in performing some of the simplest of everyday activities: such as recognizing that a colored crayon lying on the floor at the other end of the room is what is needed to complete a drawing, walking across to collect that crayon, and then putting it to use. For that matter, even the capabilities of an ant, in performing its everyday activities, would far surpass what can be achieved by the most sophisticated of today's computer control systems.*" [1]

That computers cannot compete with an ant, with its minute brain sliver, is surprising, and that they cannot do what even little children can do is obvious. The paper first argues that human and computer information processing are different designs with different strengths and weaknesses, then argues that they should therefore combine rather than compete.

2. Computer vs. human information processing

In comparing human and computer systems, the brain corresponds to a computer's central processing, not its printer or screen. The brain's trillion (10^{12}) neurons operate by electricity, and like transistors are on/off devices that allow logic gates [2]. In the neuro-computational approach [3] neural patterns encode, transform and decode information. The brain is seen as an information processor, like computer central

Figure 1. Pattern recognition

processing, but of a different type [4]. If so, what are the differences?

In a systems theory approach [5] a processing system (computer or brain) is presumed composed of processors (computer or cognitive) that receive input (from sensors or ports) and create output (to effectors or peripherals). While the brain's design is relatively consistent between people due to genetics, a computer's design is whatever its designers choose it to be. Here, "computer" refers to computers whose design follows Von Neumann's original architecture, which is the vast majority of computers in use today. In his original design, to ensure valid processing, Von Neumann assumed:

1. *Centralized control*: Processing directed from a central processing unit (CPU).

2. *Sequential input*: Input channels are processed in sequence.

3. *Single sourced output*: Output resources are locked for exclusive use.

4. *Location based storage*: Information is accessed by memory address.

5. *Input driven initiation*: Processing is initiated by input.

6. *Minimal self-processing:* System does not monitor or change its operations.

The above are not yes/no dichotomies but proposed dimensions with computer and brain at opposite ends, e.g. a computer "parallel" port has more bit lines than its serial port, but both are unlike the massively parallel millions of optic nerve fibers in the brain. Modern computers have dual-core chips and multi-channel processing but again this pales beside the brain's decentralization. The proposed differences between human and computer processing are differences of degree not kind, but are still major differences.

2.1. Control

Centralized control means all processing ultimately originates from and returns to a central processing unit (CPU), which may delegate work to sub-processors. Computers have a CPU for *control* reasons, so the computer always knows exactly where, in processing terms, it is up to. A disadvantage of this architecture is that if the central unit fails, the whole system also fails. On a hardware level, if the CPU stops so does the computer. On a software level, an operating system infinite processing loop means the whole system "hangs". Asking a room of people if their computer "hung" this week usually gives a show of hands,

especially for Windows users, but asking people if their brain "hung" this week is almost a non-question. The repetitive rocking of autism may involve neural loops cycling endlessly in parts of the brain, but such cases are rare. The brain's "operating system" can work continuously for over seventy years, while the Windows operating system gets "old" after only 2-3 years, and must be reinstalled.

The human brain, unlike the computer, has no clear "CPU". In its neural hierarchy lower sub-systems report to higher ones, but the highest level of brain processing, the cortex, is divided into two hemispheres that divide up the work between them, e.g. each hemisphere receives only half the visual field, with the left half from both eyes going only to the right hemisphere, which also mainly controls the left body side. Each hemisphere replicates its data to the other using the corpus callosum, a massive 800 million nerve fiber bridge, so both hemispheres "see" the entire visual field. Studies of "split-brain" patients, whose corpus callosum was surgically cut, suggest that each hemisphere can independently process input and create output, i.e. it acts like an autonomous "brain" [6]. Sub-systems for speech and memory within a hemisphere also have autonomy, as do psychomotor (cerebellum) and emotional (limbic) systems. *The alternative to centralized control is distributing control to autonomous sub-systems.*

2.2. Input

Sequential processing handles data or instructions one after another rather than simultaneously in parallel. While computers can use pipelining and hyper-threading, most computer processing is sequential due to cable and port bandwidth limits. While supercomputers have limited parallel processing, millions of human retina cells parallel process boundary contrast information before the signals leave the eye.

Parallel processing explains how the brain recognizes sentences or faces in 1/10th second, faster than most computers, yet the neuron refractory period is 1/1,000th second – a million-times slower than a typical computer event. Slow brain hardware allows for only 100 sequential steps in 1/10th second, and no program can do human pattern recognition in 100 lines.

The brain's slow components create fast responses using parallel processing, e.g. suppose Ali Baba is hiding inside one of forty jars. The sequential way to find him is for a fast slave to check jar 1, jar 2, etc. The parallel way is for 40 slow slaves to each check their jar independently, when:

"It is odds on that a machine - or organ - with sluggishly functioning components and a parallel mode

of operation would be able to thrash a computer with high speed components but a sequential mode of operation" [4]

While the brain processes retinal input in parallel computers scan screen pixels in sequence. *The alternative to fast sequential processing is massively parallel processing.*

2.3. Output

Single sourced output processing "locks" output for exclusive access by one processing system, e.g. two documents sent from different computers to a network printer at the same time come out one after the other, not interleaved, as each program gets exclusive access. Databases use exclusive control to avoid the deadly embrace of a double lock. In general, computers process *the same input* only one way, so software updates of a program or driver overwrite previous versions. However in the brain's evolution, new systems *overlay* rather than replace older ones, perhaps because older systems are:

a. *Reliable backups,* that take over if higher systems fail.

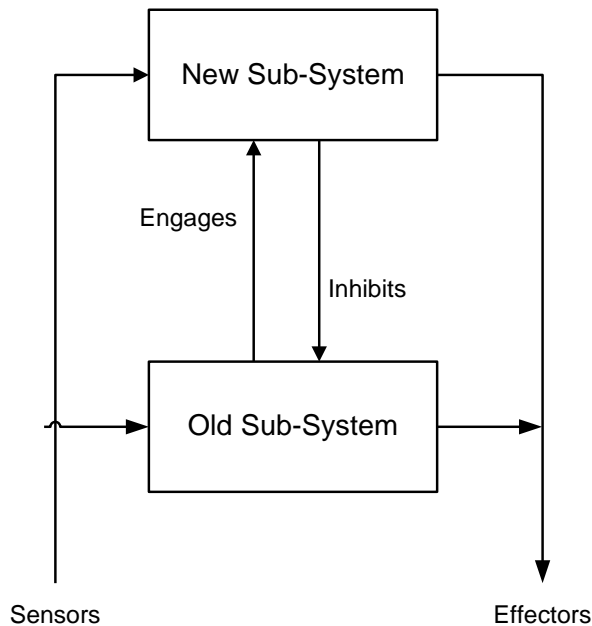b. *Faster responding*, where fast simple responses are needed.

Figure 2. Overlaid output control

Figure 2 shows a simplified view of the design where a more complex new sub-system overlays and inhibits an older one. The older system still remains and processes,

e.g. primitive brain stem reflexes that disappear in adults can reappear with brain damage. Older systems, being simpler are faster, and so can respond quicker, e.g. touching a hot stove gives an instinctive pull back. Challenger launches use three computers to calculate the complex "Launch" decision, and likewise the brain seems to calculate its outputs many ways, then take (hopefully) the best and ignore the rest. *The alternative to single source output control is overlaid output, where different systems respond on a time gradient.*

2.4. Storage

Location based storage stores and recalls information by numbered memory locations, e.g. a disk's side, track and sector. While such systems can duplicate data by duplicating storage (e.g. RAID 0), this is costly, and so one computer "fact" is usually stored in one place, so damaging the location destroys the data held there. Storage capacity depends linearly on the number of locations, so such systems can report "memory full".

In contrast, brains never report a "memory full" error, even after decades of experience. If human memory is like a data warehouse it should have a maximum capacity. If it were like a filing cabinet, specific brain damage should destroy specific information. Lashley explored this hypothesis in his well known "search for the engram" [7]. He taught rats to run a maze then surgically cut out different cortical areas in each rat. He hoped to find the maze running memory part, but found that removing *any* 10% of cortex had almost no effect, and after that maze running degraded gradually, i.e. the brain amount removed was more important than its location. The conclusion of 33 years of ablation studies was that memories are not stored in particular brain cells.

While later studies find that memory is not entirely equipotential, it is clear that one memory is not stored in one place, i.e. brains don't store memories as computers do. That electrodes stimulating certain brain cells evoke particular memories does not make them stored at that location, just activated from there. Studies suggest that one memory involves 1,000 to 1,000,000+ neurons, and one neuron contributes to many memories, rather than just one. The conclusion is that the brain somehow stores memory in the neural interconnections that increase non-linearly with neuron number, e.g. 10^{12} neurons each connected to 10,000 others gives 10^{16} connections - ample capacity for a lifetime's data. This method also allows location by content rather than address, the computer equivalent of indexing on every field in a database record, e.g. the "searches":

•   What did you eat last night?

•   When did you last have fish?

- Have you been to Auckland?
- Do you know John Davis?

and thousands of others, could all link to a single memory. *The alternative to storing information by location is storing it in the unit connections, allowing access by content.*

2.5. Initiation

The Input-Process-Output (IPO) model applies equally to the brain, whose input is the senses, and output is motor effectors.

Input driven processing means that input initiates processing which creates output, i.e. a computer's output is defined by its input. Software methods like Jackson Structured Programming use this property to deduce code from input/output specifications. If people worked this way, the brain would turn sensory input into behaviour as a mill turns flour into wheat, i.e. mechanically. Just as without wheat there is no flour, so without sensations there should be no mental processing. Yet in sensory deprivation studies people soon start to hallucinate, i.e. the brain creates perceptions. While computers without input typically fall "idle", people without input get bored, and seek out stimulation. Human processing seems not a linear Input-Process-Output (IPO) sequence (Figure 3) but a feedback loop modified by life, where output affect consequent input, e.g. turning the head affects what one sees. Psychology theory has two conflicting views on how this occurs:

a. *Objectivist*. Behaviorists like Watson, Hull and Skinner claim an objective world creates real sensations which define behavior [8].

b. *Constructivist*. Authors like Piaget, Chomsky and Maturana suggest people "construct" or interpret the world, and see *a* world not *the* world [9].
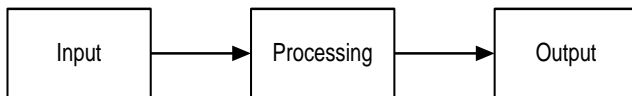
Figure 3. Linear Input/Process/Output (IPO)

Logically a feedback loop can initiate from any point, allowing an objectivist input-process-output (*IPO*) loop (Figure 4), or a constructivist process-output-input (*POI*) loop (Figure 5). Figure 4 reflects the objectivist view that input stimulus contingencies determine the system's response, i.e. the system is *input driven,* as computers are. Yet Chomsky showed it was mathematically impossible for children to learn the profundity of language in a lifetime of stimulus-response chaining, let alone a few years [10]. Figure 5

reflects the constructivist view, where genetic pre-programming of language "structures" can use the environment to develop language, i.e. human systems can be *process driven*.
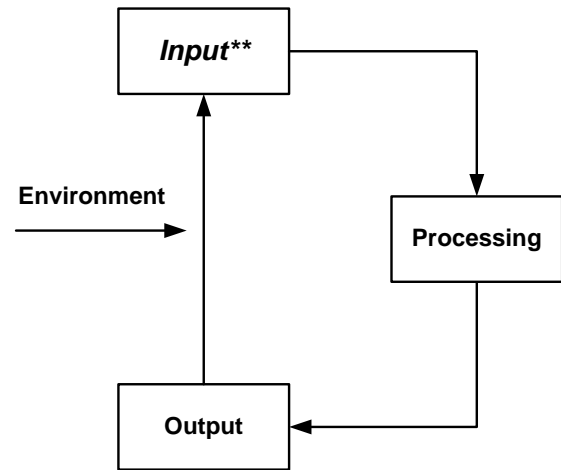
Figure 4. Input-Process-Output (IPO) feedback

Brain design allows for both models, but if anything people seem process rather than input driven, e.g. retinal signals go to the visual cortex via the lateral geniculate body (LGB) relay station, yet even more nerves go *from the visual cortex to the LGB*, i.e. in the opposite direction. Clearly the brain is not merely an input processor. In babies motor neurons develop before sensory ones, and embryos move before sensory cells are connected, i.e. output seems to precede input in phylogeny.
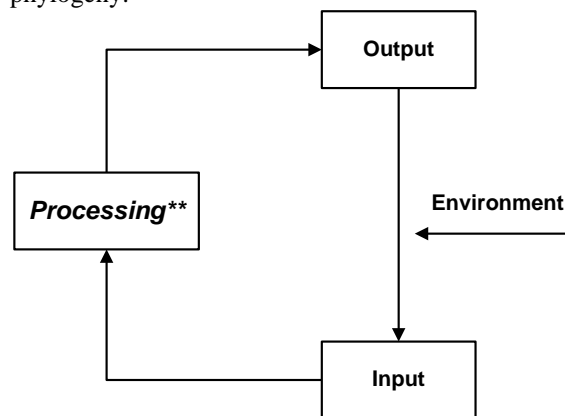
Figure 5. Process-Output-Input (POI) feedback

Simple cybernetic systems can achieve a homeostatic steady state but feed-forward systems, where output acts before input [11], can have goals, e.g. a home heating system's steady end state depends on the temperature fed into its thermostat processor. Hence in

a process-driven (POI) system the processor can begin with a temperature definition that determines the system's final end-state, i.e. the system has a "purpose". Such motivations seem to be managed in humans by the limbic system. Feed-forward loops can generate the sort of *expectation contexts* common in language, where as Gestalt psychologists noted, word meaning affects sentence meaning and sentence meaning affects word meaning, i.e. the whole affects the parts that create it. While people frequently define new contexts, computer systems are heavily context dependent, e.g. the fixed response categories of computer help (Press 1 for …).

An alternative to a mechanical input-driven system is a process-driven system that actively initiates feedback loop, allowing expectations, purposes and contexts.

2.6. Self-processing

A self processing system can process "itself", i.e. its processing loop. It is not merely that a system part inputs another's output, which is common. If a system's processing is a feedback loop, processing that processing means processing the entire loop. The classic animal test for "self-awareness" is to show a mirror to the animal and see if they know themselves. Computers, like the less intelligent animals, fail this test. In contrast, people invest a great deal of time creating their "ego", or idea of themselves, which self-concept strongly affects behavior. While computers do not have an "I", or give themselves names, most people do. While programs generally avoid writing over their own code, and an operating system that overwrites itself is considered faulty, human goals like: "To be less selfish" imply people changing their own neural "programming".

Is a "self" processing itself, like a finger pointing to itself, impossible? Not for overlaid systems, as brain systems like the frontal cortex, with autonomy, can observe the rest in action, and so form not only a concept of self, but also use it in social relationships. Social Identity Theory further suggests that groups arise when individuals use a group's "identity" to form their self identity [12]. The normative behavior groups use to cohese can be linked to the concept of self [13], i.e. social development requires the ability to self-process.

The alternative to simple processing of external data is for one autonomous system part to process the processing of another, allowing self-awareness, ideas of "self", social interaction and self-change.

2.7. Summary

In summary, the brain's design differs from that of most computers in its:

1. *Decentralized control:* The brain has no CPU, even at the highest level.

2. *Massively parallel input:* Retinal cells are massively parallel processors.

3. *Multi-sourced output*: New and old brain systems compete for output.

4. *Storage by connections*: Allows access by content and lifetime's memory storage.

5. *Process driven initiation*: People plan, expect, hypothesize and predict life.

6. *Self-processing*: People have concepts of self and group that allow social activity.

The human answer to information processing is the computer, a powerful system with one central control point, that processes most input signals one at a time, that uses one program at a time for output tasks, that stores one bit of data in one place, that initiates its processing in one way, and that does not process its own processing. In contrast, nature's solution is more subtle, with many points of control, many input signals processed at once, many output calculations per task, memories stored in many places, initiation by processing as well as input, and meta-levels of processing. Such design differences suggest that machine intelligence and human intelligence may be as different as apples and oranges. The brain uses tactics that for computers are highly risky, e.g. self-processing risks infinite recursive loops. Yet risk also enables opportunity, so people unlike computers can think about their thinking. While computers have people to look after them, the brain responds to undefined and potentially infinite variability in real time, where "It does not compute" is not an option.

3. Implications

The brain as a different type of information processor has implications for computing development.

3.1. When smart computers turn dumb

Ever since an automated computer billing system posted a one cent invoice in a 20c envelope people have wondered how smart computers really are. As computers attempt human specialties, like walking and talking, they tend to look stupid rather than smart, e.g. attempts to "converse" with automated artificial intelligence (AI) help so frustrated users that phone help providers now typically only use AI initially, before giving a human help, if at all. Many users bypass the computer choices by immediately pressing 0 to get an operator. The vision of fully automated interactive AI help has not eventuated as expected. As we get to know

computers, their image of incredible cleverness is changing, e.g. AI enemies in computer games are considered less challenging than human opponents, hence the rise of multi-player online gaming.

While the initial "power" advances of AI were rapid, it seems now to have struck what can be called *the 99% barrier*, e.g. computer voice recognition is currently 99% accurate, but one error per 100 conversation words is an error per minute, which is inadequate. There are no computer controlled "auto-drive" cars because 99% accuracy means an accident every day, again unacceptable, e.g. of the 23 autonomous ground vehicles that started the DARPA Grand Challenge 2005, only five finished [14]. Real world requirements, like driving in rush hour traffic, need well above 99% driving accuracy. A competent human driver's "mean time between accidents" is in years, not days or months, and good drivers go 10+ years with no accidents.

Human information processing has somehow crossed over into "the last percentage" of performance, but for computers more power is now giving diminishing returns, perhaps because processing power alone is not enough for "equivocal" real world tasks [4], e.g. pure processing logic cannot deduce a 3D world from two-dimensional retinal signals, as the brain does. David Marr suggests that computer pixel-by-pixel processing has not lead to face recognition because trying to understand perception by studying neuronal (pixel level) choices is "*like trying to understand bird flight by studying only feathers. It just cannot be done.*" [15]. The solution may require not just processing power but a new processing style. Yet if computers deviate from von-Neumann's original deterministic assumptions, who is to say they will not then inherit human-like weaknesses?

Despite enthusiastic claims that computers will soon overtake people in intelligence [16] technology still struggles to simulate retinal activity, let alone the visual cortex [17], i.e. computers struggle to simulate the lowest level of brain processing (the retina is part of the brain). The problems of say computer-vision are only just beginning to be appreciated:

"Computers are no real competition for the human brain in areas such as vision, hearing, pattern recognition and learning. ... And when it comes to operational efficiency there is no contest at all. A typical room-size supercomputer weights roughly 1,000 times more, occupies 10,000 times more space and consumes a millionfold more power ..." [17]

Barring an unexpected breakthrough, computers in the foreseeable future will struggle with skills like conversation that five year olds find trivial, raising the question: "*How long before computers learn what people learn after five?*" Humans at 18 years are considered "immature" not only for complex social tasks like managing, but even for physical team sports like soccer. The Robot World Cup (http://www.robocup.org) wants robot teams to compete with people by 2050, but the contrast between robot shuffles and world cup brilliance may be more than imagined. Perhaps the question is not whether 50 years will suffice, but whether a thousand will.

Consider the possibility that nature tried the simple power approach of computers today, found it wanting, and moved on, e.g. Kim Peek, who inspired the movie Rain Man, is developmentally disabled (he has no corpus callosum) yet knows by heart every word on every page of over 9,000 books [18]. Savant syndrome, which generally occurs in people with IQs of 40-70, often from left hemisphere damage, gives people who can calculate 20 digit prime numbers in their head yet need social care to survive. Modern computers, with their amazing abilities, may be the electronic equivalent.

3.2. A socio-technical approach

The socio-technical approach gives a pragmatic model for computing progress that does not depend upon computers outperforming people in real life tasks like driving. In this view, systems operate on four levels [19]:

1. *Mechanical/Physical.* *Computer hardware, wires, printer, keyboard, mouse*

2. *Data/Informational.* *Software programs, data, bandwidth, memory, processing*

3. *Social/Human.* *Human meaning, semantics, attitudes, beliefs, opinions, ideas*

4. *Social/Group.* *Group norms, culture, laws, zeitgeist, sanctions, roles*

These levels match the definition of IS/IT as hardware, software, people and business processes [20], and fit the concepts of Kuutti [21] and Grudin [22]. Computers are at the lowest level hardware chips and circuits, yet software "emerges" from hardware as a new system level, based on data flow diagrams not circuit diagrams. Likewise meaning can "emerge" from data/information, and cultures from individual meanings. Higher levels offer higher performance based on higher operational requirements, e.g. social synergy requires social justice.

Figure 6. Utopian technology progress

Technology Progress
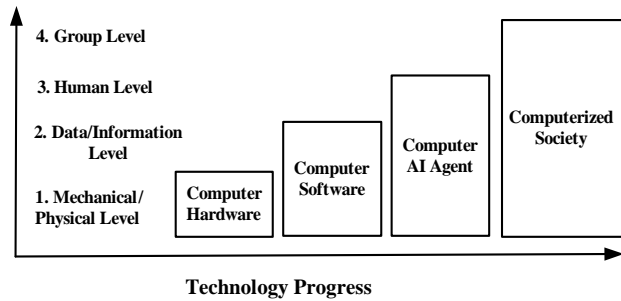
In this view (Figure 6), computers will progress beyond mere information processing, into areas like thinking and socializing, and soon overtake human processing, if they have not done so already. The problem with this technological utopianism is that computer AI agent development has made little progress in a decade.

In the alternative socio-technical model (Figure 7), computers combine with people to form new systems, e.g. while plane and pilot can be seen as two physical systems (human and machine) working side-by-side, or seen as *one socio-technical system with human and mechanical levels* (where the pilot's body is just as physical as the plane). In the latter case the human adds a new system level not just a new system component. In a socio-technical view, computers need not attempt tasks like abstract thinking that a perfectly good information processor (the brain) already does, perhaps as well as can be expected given the task. If people and computers are simply different types of information processors, this approach says "vive la difference" and aims to combine their strengths. The goal of computing now changes, from making better computers to forming better human-computer teams, i.e. from computer excellence to human-computer excellence. The computer's role also changes, from that of clever independent actor to human assistant [23] or social environment.
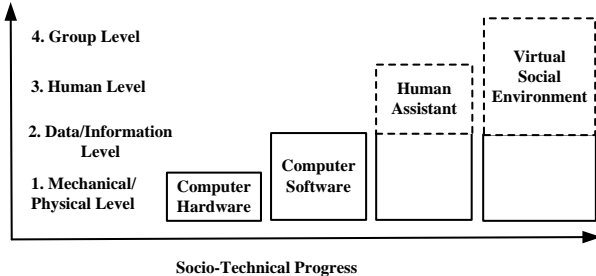

Socio-Technical Progress

Figure 7. Socio-technical Progress

In Figure 7, the computer progression into human and social realms is shown dotted, to indicate it is achieved in cooperation with people. Also, socio-technical development is seen as an extension of HCI concepts.

To support the view that people plus computers are more powerful that people or computers alone, note that every runaway IS/IT success of the last decade has supported rather than supplanted human activity. The "killer applications" of Table 1 support well known human processes rather than trying to take them over, e.g. email systems display and transmit messages but leave creating and reading them to human senders and receivers. Computers that combine human and computer processing are succeeding more than clever systems with stand-alone processing, like the Sony dog. A Sony dog with less smarts but cuddly fur and puppy dog eyes might be more successful. The principle of finding a human task and supporting it has many success stories, e.g. tax software, route direction systems (Mapquest), currency conversion, etc.

While driverless cars are still a distant possibility, automobiles already have features like reactive cruise control, range sensing and assisted parallel parking [14]. While computer surgery is still a dream, computer supported surgery (e.g. over distance) is a reality. While robots routinely fall over, people with robotic limbs walk well. However designers now have a problem beyond computer excellence, namely defining the computer/human boundary.

Clever software that crosses that computer-human boundary and attempts human roles becomes annoying not useful. Word can be like a magic world where moved figures and titles jump about or disappear, typing "*i* = 0" turns into "*I* = 0", tables refuse to resize their columns, and text blocks suddenly change to a new format (like numbered) after a deletion. Experienced users typically turn off auto-help options as, like the sorcerer's apprentice, software assistants soon get out of control and act beyond their ability. The saving grace is that Ctrl-Z (Undo) can usually fix clever software's errors, but this is not always the case, e.g. Endnote X "lost" all the reference links for this paper, probably because it was written on two machines (home and work), one with an empty reference database. Whereas early "dumb" versions of this software only updated references when asked, the latest "smart" version does it without asking, and in this case, unaware it was outside its assumed context, replaced valid data fields with null ones. Software that excludes the user from its operation then acts beyond its ability is an increasing problem, not counting the programmer time wasted on functions that users turn off like Mr. Clippy

Table 1. Human processes and killer applications

Application	Human/Social Process
Email	Conversation
Blogs	Expression
Wikipedia	Sharing knowledge
E-bay	Trading
Hypertext	Associative thinking
Reputation systems	Group cohesion
Social Networks	Making friends
Chat	Group conversations
Browser	Information gathering
Online games	Playing games

Rather than software using hidden logic to predict what users want before they know it, why not watch the choices people make *and follow that direction?* If a good assistant watches and remembers, Word is not very good, e.g. I repeatedly change Word's numbered paragraph default indents to my preferences, but it never remembers them. Its programmers are either unable or unwilling to replace its defaults with mine. It cannot follow what I do, but requires every instruction be given literally. Even worse, it acts like it knows better, e.g. if I ungroup and regroup any figure it takes the opportunity to reset my sensible options on say text wrap-around to its inappropriate defaults, so the picture now overlaps the text. The software "sees" the document but seems only dimly aware of the user's interaction with it, e.g. it stores innumerable pointless document details, but cannot remember where I last put the cursor (and put me back there when I re-open the document). The "format paint" function is useful because it lets users tell the computer "Do as I just did". Likewise most people use Word's list of last opened documents, but for some reason it limits this to 9 instead of say 20, and users cant "mark" a document to keep on the list. Computing needs to change its image from upstart leader to simple assistant, as competent users dislike software that is too clever for its own good and won't listen.

3.3. Humanizing technology

For a computer system to support human requirements they must be specified, i.e. computer designers must analyze human processes to design a technology fit. Interfaces that work the way people work are more accepted, more effective and easier to learn. The earlier processing differences suggest the following interface requirements, some of which are already met but others are not:

1) **Manage user attention**. If the brain is a collection of autonomous sub-systems that distribute control in a "Society of Mind" [24] type market place, then "attention" is when one neural sub-system "purchases" control, either by the speed or intensity of its response. This suggests two effects:

Concentration: Higher systems exert *top-down control over* lower sub-systems.

Distraction: Lower systems use *bottom-up control* to engage higher sub-systems.

Which effect is good or bad depends on the context, e.g. a colored "New" graphic at the start of a text sentence may usefully "distract" a user to begin to read it, but the same flashing graphic at the end of a sentence makes it difficult to read (as while reading one is continuously distracted to the flashing at the end). *A good interface manages attention, as the entire screen is not processed in the user's scan.*

2) **Engage available user input channels**. If human input involves parallel channels, computers should use those channels using multi-media web sites. However multi-media really means multi-channel, as text meaning, depth, texture, color, shape, movement and orientation are distinct channels but the same visual medium. These human processing channels are "always on", so adding them does not increase information overload, e.g. a web site with no depth cues merely leaves visual depth processors idle and reduces the user experience. Adding a screen background takes no more effort as users have processors dedicated to handle background textons. Multi-channel input gives interface designers something for nothing. *A good interface engages the many channels of human neural processing.*

3) **Support multi-level responding**: If people have both primitive (fast) and sophisticated (slow) responses, an HCI interface should handle both, e.g. users may assess a web site instantly on its "feel" and later decide its value. Designers must satisfy both demands, as a useful interface that fails quick analysis may not reach the second stage, as the user has already clicked on. *A good interface meets both immediate impressions like boundary contrast, and long term impressions, like understandability.*

4) **Support knowledge associations**. If people access knowledge by associations, it makes sense to design interfaces that access information the same way, e.g. hypertext links let people recall by connecting from one knowledge element to the next, linking words within a document and words to other documents. People recall by connecting rather than retrieving. Hypertext has succeeded because it works as human memory works, where one thing leads to the next. *Good interfaces let*

users remember by associations, rather than demanding full information up front.

5) Provide feedback flow. Even a multi-media visual feast is boring if one is passive. Interactive web sites are interesting because people act on them and feel in charge. The Back button was a great software invention as it gave interface control to the user. Students who struggle to spend an hour passively "inputting" textbook data can spend 4 hours/night battling imaginary enemies in a game because games give feedback and books don't. Actively driving a feedback loop is naturally rewarding. The three click rule illustrates the feedback demand that a response occur at least every three clicks, and progress bars also keep the feedback loop turning. *A good interface gives users regular feedback flow.*

6) Remember social interactions. Self processing allows the concept of self from which social relationships evolve. However typically computers socially blind, e.g. Mr. Clippy, Office '97's assistant, was a paper clip figure who popped up to ask 'Do you want to write a letter?" any time one wrote the word 'Dear". Using advanced Bayesian logic, he was touted as the future of "smart help", yet a PC Magazine survey found him the third biggest software flop of the year [25]. The problem was that while Mr. Clippy analyzed *your* document actions, he had no memory whatsoever of *his own interactions with the user*. No matter how many times one told him go away, he happily popped up next time offering to "help", like a friendly drunk who will not leave. People expect "smart" software to be smart enough to recognize rejection. Mr. Clippy, like most ''intelligent'' software today, was blind to social interaction. *A good interface does not interrupt or annoy and remembers past user interactions.*

The approach is to *derive computer primitives from psychological processes,* e.g. multi-media systems succeed by matching the many human senses. HCI analysis then involves:

1. Specify the overall task, e.g. human conversation

2. Allocate the human role, e.g. create/read meaning

3. Design the computer system support, e.g. e-mail.

3.4. Socializing technology

The nature of virtual communities is beyond the scope of this paper. However for technology to succeed as *a social medium* it must address social issues like fairness, copyright and privacy. Without justice for example, any society, whether physical or electronic, tends to become unstable. The social level of computer analysis considers issues of "rights", i.e. who can do

what to what in the electronic environment, e.g. spam arises when email gives sender's rights over receivers, and online copyright is an issue because browsers give viewers more rights than those who post information on the internet [26].

Again the general approach is to *derive computer primitives and applications from social processes,* e.g. E-bay's rules are similar to those of other markets. This principle can create new applications, e.g. normative behavior, where people "conform" to the group they belong to, is a well known social effect. It applies when people look in restaurants not to see the menu, but to see how many people are eating inside. Such "herd" behavior has problems like "groupthink", yet it is often efficient to follow group knowledge, e.g. people often traverse a forest by following the paths trod by others. Web sites could support this social process by showing "web-tracks", i.e. changing link appearance to reflect use frequency, e.g. often-used links could increase in size or deepen in color. Web "tracks", that show visually where other users have gone, would be highly successful. Web-trackers already gather user click data in secret for web site owner benefit. A web track site would simply make click data visible to everyone, not just a few. Of course for web-tracks to work as a social tool, social problems would have to be overcome, e.g. a social standard would be needed to prevent sellers from abusing it to trick customers.

4. Conclusions

It is concluded that:

1. People and computers are different types of processors, with different strengths and weaknesses

2. For computing to attempt human specialties may be not smart but dumb.

3. Computing that works with people and society will succeed more than that which tries to advance alone.

4. Human and social processes will increasingly drive computer design.

The next decade will show whether these general predictions are true or not. Perhaps in a few years robot house-help will walk in the door. Even so, the argument is not against technology-centred progress, but for socio-technical combinations, i.e. why waste the processing power of the human brain? It asks designers to ask how users can help as well as hinder. Indeed the principle that higher performance requires higher system levels suggests that for a global society to arise from Internet technology it must follow rather than ignore human social guidelines.

References

[1]   R. Penrose, *Shadows of the Mind*. Oxford: Oxford University Press, 1994.

[2]   W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics,* vol. 5, pp. 115-33, 1943.

[3]   P. S. Churchland and T. Sejnowski, *The Computational Brain*. Cambridge, MA: MIT Press, 1992.

[4]   J. Copeland, *Artificial Intelligence*. Oxford: Blackwell Publishers, 1993.

[5]   L. v. Bertalanffy, *General System Theory*. New York: George Braziller Inc, 1968.

[6]   R. W. Sperry and M. S. Gazzaniga, "Language following surgical disconnexion of the hemispheres," in *Brain Mechanisms Underlying Speech and Language*, C. H. Millikan and F. L. Darley, Eds. USA: Grune & Stratton, 1967.

[7]   K. S. Lashley, *Brain Mechanisms and Intelligence*. New York: Chicago University Press, 1929.

[8]   B. F. Skinner, "'Superstition' in the pigeon," *Journal of Experimental Psychology,* vol. 38, pp. 168-172, 1948.

[9]   H. R. Maturana and F. J. Varela, *The Tree of Knowledge*. Boston: Shambala, 1998.

[10]   N. Chomsky, *Language and Mind*, Third ed. Cambridge: Cambridge University Press, 2006.

[11]   O. Mayr, *The Origins of Feedback Control*. Cambridge: MIT Press, 1970.

[12]   M. A. Hogg, *Social Identity Theory*: Springer-Verlag New York, 1990.

[13]   B. Whitworth, B. Gallupe, and R. McQueen, "Generating agreement in computer-mediated groups," *Small Group Research,* vol. 32, pp. 621-661, 2001.

[14]   I. Miller, E. Garcia, and M. Campbell, "To Drive is Human," *Computer,* vol. December, pp. 52-56, 2006.

[15]   D. Marr, *Vision*. New York: W. H. Freeman, 1982.

[16]   R. Kurzweil, *The Age of Spiritual Machines*. Toronto: Penguin Books, 1999.

[17]   K. Boahen, "Neuromorphic Microchips," *Scientific American,* vol. May, pp. 56-63, 2005.

[18]   D. A. Treffert and D. D. Christensen, "Inside the Mind of a Savant," *Scientific American,* vol. December, pp. 108-113, 2005.

[19]   B. Whitworth, J. Fjermestad, and E. Mahinda, "The Web of System Performance: A multi-goal model of information system performance," *Communications of the ACM,* vol. 49, May, pp. 93-99, 2006.

[20]   S. Alter, "A general, yet useful theory of information systems," *Communications of the AIS,* vol. 1, pp. 13-60, 1999.

[21]   K. Kuutti, "Activity Theory as a Potential Framework for Human Computer Interaction Research," in *Context and Consciousness: Activity Theory and Human-Computer Interaction*, B. A. Nardi, Ed. Cambridge, Massachusetts: The MIT Press, 1996.

[22]   J. Grudin, " The Computer Reaches Out: The historical continuity of user interface design.," in *Proceedings of CHI '90, ACM SIGCHI Conference*, Seattle, Wash., USA, 1990.

[23]   C. A. Miller, "Human-Computer Etiquette: Managing expectations with intentional agents," *Communications of the ACM,* vol. 47, pp. 31-34, 2004.

[24]   M. L. Minsky, *The Society of Mind*. New York: Simon and Schuster, 1986.

[25]   PCMagazine, "20th Anniversary of the PC Survey Results." vol. 2004, 2001.

[26]   B. Whitworth and E. Whitworth, "Reducing spam by closing the social-technical gap," *Computer,* pp. 38-45, 2004.