Chapter 4

DIGITAL SYSTEM ARCHITECTURE

Probably the likeness between the digital paradigm and pharmacology, and the similarity between the analog paradigm and herbalism appear rather reasonable after the first pages. However I have not examined as yet what happens in the working environment; I have not observed carefully what digital experts do when they design a device, when they build up an infrastructure or optimize a medium. Electronic engineers create a broad assortment of solutions and the reader perhaps wonders whether they are guided by rational criteria also inside factories and laboratories; and whether they use theoretical models connected to the notions of signifier and signified discussed above. The resemblance of digital experts with chemists should be validated in the professional practice, when customers demand and pay for an appliance or a service.

In the next pages I aim at an exploration of the living environment and shall discuss the following topics:

a) *The hardware architecture of digital systems* – Chapter 4,
b) *Computer networks* – Chapter 5,
c) *Storage* – Chapter 6,
d) *Protection/security measures* – Chapter 7,
e) *Software programming* – Chapter 9, 10.

Speaking in general, the principles of a discipline are to explain the basic aspects of this discipline. This is the first task to accomplish, thus I shall comment on the essential features of the points from *a*) to *e*), and shall overlook the specialists' details.

I shall proceed in the following way: I shall first present an inference derived from the contents of the preceding pages in the abstract, namely I shall deduce a criterion of work consistent with the present logical framework. In the second stage we shall check whether the criterion of work defined on the paper has been adopted by digital experts in professional practice.

1. FROM INFORMATION TO UNITS

Digital systems are required to manipulate a broad variety of information: numbers, as well as texts, pictures, graphics, music, symbols, and sounds. Those systems achieve mathematical operations and graphical elaboration, they assemble, store and correlate messages and execute several algorithms. Because of such an ample assortment of services, I am inclined to conclude that the hardware architecture of a digital system exploits the properties of information in a systematic manner.

I assumed that the internal of the machine *S* depends on the qualities of the product *w* that *S* brings forth; thus the following features should determine the operations of a computing machine:

1. *A signifier has a physical basis*,
2. *A signifier stands for something*.

I mean to say that a digital system should transform *E* and *NE* respectively; it should fully exploit properties (1) and (2) in order to process information without restrictions. I proceed by deductive reasoning and conclude that a system should achieve two families of processes determined by (1) and (2), and this pair of functions should have the following profiles:

[1] The first process should consist in changing the concrete form of input data. This operation transmutes the signifiers and keeps the meanings of input and output unaffected.

[2] The second process should output novel semantic contents with respect to the input. This function elaborates the subject matter of the input messages and delivers original pieces of news through an automatic process while the physical appearance of signifiers steadies.

Are these conclusions true in the real world?

A. Digital Systems

At the present time, electronic manufacturers offer instruments of measurement, control units, optical devices, medical equipment, calculators and many other original implements in digital technology. Universal experience shows how all those devices are equipped with *converters* that achieve function [1] and with *electronic chips* that bring forth function [2]. For instance, a cell phone includes integrated circuits that process signals and four converters: a microphone, a speaker, a keyboard, and a screen. Thus the special method of study which I introduced in the first pages proves to be correct. Concrete evidence supports the reasoning I have just developed.

However, I should improve the validation of my theory and the incredible variety of digital solutions interferes with the accurate analysis of those machines. I am running into some trouble but happily find assistance in the technical terminology which places all the digital devices into two boxes. An electronic appliance falls either in the group of *special* *purpose* (*SP*) *digital systems* or in the group of *general purpose* (*GP*) *digital systems*. The former is a machine specialized in a definite number of functions, e.g. a digital camera and a digital control station. The latter is a flexible machine that can fulfill different duties and runs in various environments e.g. a personal computer and a tablet.

The *conversion functions* [1] and *data-processing functions* [2] are not so visible in a SP system. Things are better in a GP computer. The physical parts [1] and [2] are so large as to constitute independent units. For example a printer – which converts electrical impulses into signs of ink – can be separately purchased. Moreover a GP system executes a broad variety of functions, hence the discussion of conversion and data-processing operations will preferably refer to computer systems in the next pages.

A. Converters

Computers handle physical parameters such as temperature, hours, and pressure that are to be displayed in readable forms. Facts show how a GP system operates with a variety of signifiers required by users and environmental entities, and converters cross certain types of boundaries from continuous to discrete, from a physical form to another physical form and so forth.

The technical literature places the conversion devices into the ensuing classes:

1) *Sensors* transmute a physical signal into another physical signal for example an optical signal into electrical;
2) *Actuators* are used to transform a signal into a motion. They are typical components of robots;
3) *Analog-to-digital converters* (ADC) change the form of an electric signal from digital to analog (and vice versa);
4) *Modulators* change square waves into continuous waves and vice versa;
5) *Transcoders* change codewords from one coding system to another for example from ASCII code to EBCIDIC code.

I mean to overlook appliances 3, 4 and 5 since the most significant converters transmute the physical nature of signifiers and belong to classes 1 and 2 which have been inventoried in the previous chapter. LED and LASCR have been quoted as examples of optical converters; a key in the keyboard offers a third straightforward case. When you press the key *A* on the keyboard, you enter a piece of information which is mechanical. The key is a switch that receives a mechanical signal and emits a sequence of electric impulses, namely it converts the material nature of *A* from mechanical to electrical.

Only an analog appliance can handle natural signifiers such as the finger press, colored light, the rotation of a body etc. and the technical literature officially classifies sensors and actuators as analog equipment (see previous chapter). A device 1 or 2 can include digital circuits, but those circuits work for the analog core which transmutes the signals. Conversion operations are under the charge of analog components.

Converters come in many dissimilar types due to the variety of features of the phenomena they measure: how they work, what they are made of, their function, their cost and their accuracy; it turns out that two devices belonging to the same class can have far differing shapes, volumes and prices. We find a converter as small as a tiny chip and a converter as huge as a unit equipped with sophisticated electronic circuits. Printers, display screens, loudspeakers, microphones, keyboards, and faxes are popular conversion units that exhibit different appearances and dimensions (Doyle 1999).

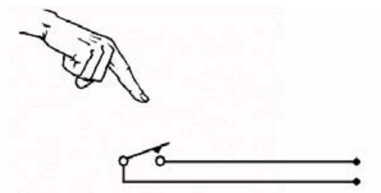

Figure 4.1. Mechanical-electrical converter.

Companies commit substantial investments in research and development of new transducers for printers that change electric impulses into ink letters on the sheet; for display screens that convert electric bits into bright points; for loudspeakers that transmute electric waves into sounds; for disk-drivers that switches magnetic bits into electric impulses. Charge coupled devices (CCD), LED, magnets, electric motors and other parts provide examples of analog components employed for conversion purposes and demonstrate how statement [1] comes true in the world.

B. Data-Processing

Let us conduct an accurate examination of what data-processing really consists.

1. Numerical Data-Processing

We begin by scrutinizing *numerical data-processing* which probably is the most familiar form of data-processing. Take the following subtraction where the input numbers signify the price of a product, and the payment by cash

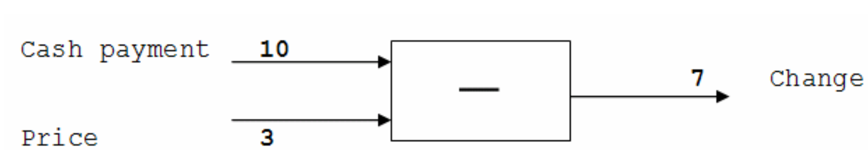

Figure 4.2. Numeric processing of data.

The output number of this subtraction stands for the change that the cashier has to give to the client. Note how the outcome symbolizes something new with respect to *10* and *3* in input and for this reason *7* is useful to the cashier. The result of data-processing conveys meanings quite different from the input.

This case appears rather elementary as an electrical device creates novel pieces of news through several operations. A satellite which flies around the Earth suggests a more complex case to examine. The speed of the satellite is obtained by the time derivative of the space function $s(t)$ and the acceleration is given by the second derivative of $s(t)$

$$v(t) = \frac{ds(t)}{dt} \qquad a(t) = \frac{d^2 s(t)}{dt^2}$$

The control station calculates the derivatives, and minute by minute exhibits the results on the control panel to flight controllers. Numerical data-processing proves to be precious because in advance of computation the speed $v(t)$ and the acceleration $a(t)$ are unknown pieces of information for the flight controllers.

Numerical data-processing matches with definition [2] instead the abstract study of Computing hints at the idea that the systems handle exclusively *abstract numbers* – see Chapter 1, Paragraph 5 – and therefore prevents people from grasping definition [2]. I am not sure that one can realize in what processing of data consists, in reality as long as he/she discards the significance of numbers.

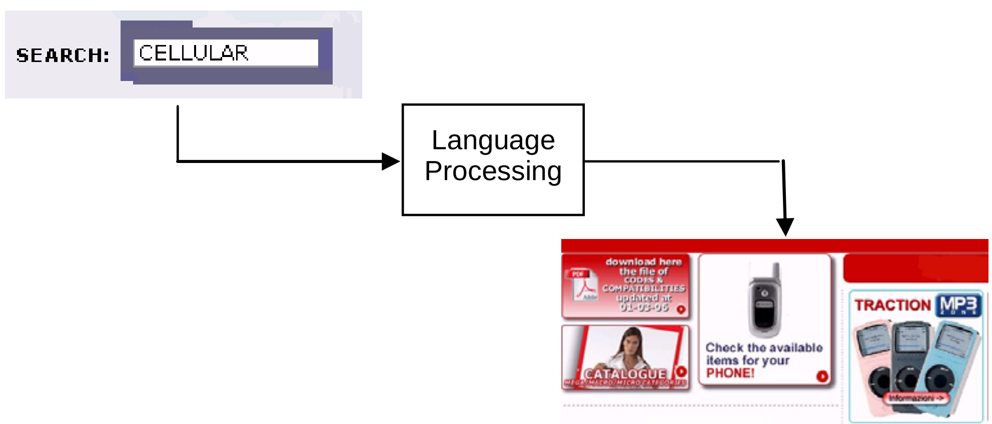

Figure 4.3. Linguistic processing of data.

2. Verbal Data-Processing

Probably the reader has read a sentence of this kind: "Data-processing consists of automatic mathematical calculations". This definition is to a certain extent true but does not fit with the variety of operations executed by modern systems. Besides numerical data-processing more forms of data-processing are widely in use.

All the day long modern computers process linguistic information. Suppose you press the key-word CELLULAR in a search engine, and in a few seconds you obtain several news and offerings from the Web.

You can now purchase mobile phones at low cost; in fact the system has provided you with messages whose contents absolutely differ from the input key-word.

3. Visual Data-Processing

Suppose that a satellite takes a snap and a special coloring program highlights the area rich in mineral resources using false colors. The announcement of a bonanza is the final message ignored in advance of *visual data-processing*.

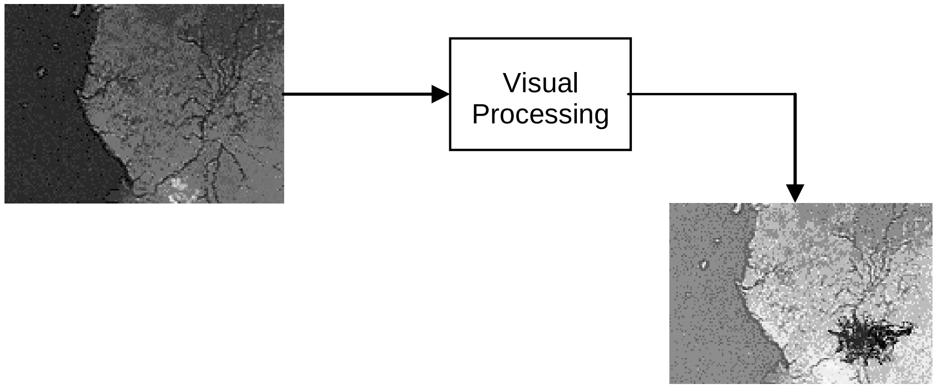

Figure 4.4. Visual processing of data.

4. Operational Data-Processing

The responses of data-processing just seen are symbols, images etc.; the output may even be an operational signal that guides a device. For example the subtraction in Figure 4.2 can control a mechanical cashier that gives the change to a customer through an automatic drive.

The digital system S that has the control of the device C offers another example. Suppose C breaks down and emits the bit string "10011" to S, promptly the system response "01101" switches off the device. The digital system sends a signal whose meaning differs from the input meaning, that is to say, S executes an *operational data-processing*.

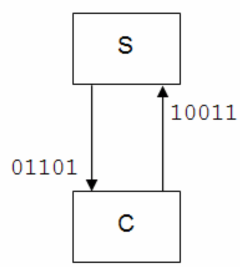

Figure 4.5. Data-processing for control purposes.

The meanings NE_{input} of "10011" and NE_{output} of "01101" are independent of human feeling. It is unnecessary to translate NE_{input} into English words such as "failure" and NE_{output} into "switch off". The verbal descriptions are superfluous since the strings "10011" and "01101" determine two specific actions. It may be said that "10011" and "01101" have proper significance in accordance with the remarks placed in the last part of Chapter 1.

5. Static and Dynamical Responses

Data-processing provides single data items or otherwise provides dynamic results and portrays a real event, or a state of affairs. When the system displays realistic images and/or allows the operator to use realistic devices, then we have a *simulation*. For example, a flight-simulator immerses the operator in a virtual world and the operator has the impression of flying anywhere in the world by means of a joystick.

Figure 4.6. Flight simulator.

Simulation is used in many contexts, including Physics, Chemistry and Biology as well as engineering and even Economics and Social Science, for the purpose of gaining information about phenomena which will occur or have occurred in a mysterious way.

Practical evidence shows how data-processing transforms the input subject matters and prepares novel pieces of news; it delivers original messages; it makes unexpected representations of the reality and conforms to definition [2]. The present interpretation of data-processing based on semiotic notions is consistent with common experience.

In current literature some authors talk about "*transformation of content*" and mean to describe the translation of a text into a more useful format or into a more meaningful structure. This operation occurs especially in the Web where different platforms and systems communicate (Stojanovic 2009) but this special transformation of contents has nothing to do with the present argument.

6. Impact of Data-Processing in the World

It may be said that most economic processes are information-based in the modern world. Managers and leaders make influential decisions with success as long as they have trusty elements in hand, but the manual acquisition of information normally requires exhausting efforts and substantial investments. The possibility of obtaining pieces of news via swift mechanical processes turns out to be an extraordinary opportunity. The automatic production of information items has a vital value and a great effect in the civilized world and in emerging economies.

The posited concept of data-processing [2], which is grounded on Semantics, offers an aid to explaining why computers support people everywhere, how computers assist individuals and communities and achieve dominant success. Conversely one is inclined to underestimate the relevance of systems to modern economies when he/she assumes that digital systems treat ethereal values. Computers have a considerable influence on our lives and strongly affect the course of events and human relations because the strings of bits represent dollars, euro, gallons, products etc. If by absurd, computing machines ran exclusively with abstract data, they would have a negligible impact on real life. Very few people would pay attention to data that has generic reference to physical reality. The extraordinary contribution of data-processing evaporates and the importance of computer systems vanishes in the mind of those who have some sort of ethereal view of Computing.

The abstract approach to Computing prevails in some scientific groups which overlooks the semantic aspects of data and believe that data-processing coincides with calculus. This idea was not completely wrong in the pioneering age; most software programs were mathematical programs in those years. Things no longer go in the same way.

Nowadays a lot of programs handle sounds, texts, letters, and pictures which have nothing to do with numbers. Surveys conducted on the use of programming languages reveal that mathematical computations constitute a minority group in comparison with the other forms of data-processing. Fortran ranks at the last place of popularity for PC programmers (Norton et al 2002). Mathematics-oriented languages such as Fortran and APL hover around 5-10% of programmers' work-time in mainframe environment. Other recent statistical results imply that currently numerical data-processing is not a large portion of computer applications. Innovative data treatments – linguistic, visual, multimedia etc. – are overwhelming and the percentage of mathematical programs is decreasing day by day.

All this should help the reader to gain and to consolidate a practical view of data-processing.

7. Creative

Data-processing – the most astonishing activity of computers – brings into existence original pieces of news through mechanical rules. It is not an exaggeration to claim that data-processing simulates human thought, in fact thinking is the improvisation of something which has never existed before. Creative reasoning is the process which people use when they come up with a new idea. And the information processes illustrated from Subsection 1 to 5 in this paragraph exhibit something like creative thinking.

Since the seminal article by Turing who introduced the '*test for consciousness*', scholars argue about the possibility of reasoning for computers and conduct endless debates on whether a machine can discover something new (Koch et al 2008).

The controversy centers upon two elements: the computing machines and the human mind. The former appears rather complex and the latter really constitutes a knotty conundrum. The two obscure terms of the problems – the computer and the brain – need to be clarified, and the present logical framework offers a small contribution toward that goal. The proposed definition of data-processing casts light on the *creativity/ inventiveness* of machines.

From the present perspective, it is evident that data-processing proves to be a *creative* process as long as the computing machine generates original pieces of news. Though data-processing does not constitute an *inventive* process in that digital systems apply the procedures assigned by programmers who establish the plan of work through software instructions. A processor does not concoct an answer because it cannot operate on the basis of an original idea as man does. Intuition does not trigger an appliance that instead rigidly executes the scheduled operations even in the most advanced applications in Artificial Intelligence.

8. How Data-Processing Comes to Be True

Perhaps the creativeness of data-processing puzzles the reader who wonders how data-processing works in reality. The doubt may be expressed in the following terms:

How can a computer system convey novel information if that machine is unconscious, if circuits are unaware of data meanings?

Usually authors view this argument through philosophical glasses. Thinkers debate the autonomy and the consciousness of machines; instead I like better to dissect the material elements of digital systems that carry on data-processing.

Mechanical data-processing brings forth novel messages due to complex sets of *operations* coordinated through cells of memory called *data-fields* (and also *areas*, *elements*, *spaces* etc.). Computer operations handle strings of bits in various manners with the support of data-fields that are portions of space which keep pieces of information for a short while (Kraft 1979) (Scott 2008). In substance, a specialist – either hardware or software – designs and implements automatic data-processing by means of correlated instructions and data-fields. Thanks to this couple of tools, a specialist commands a machine to bring forth novel contents through an automatic process.

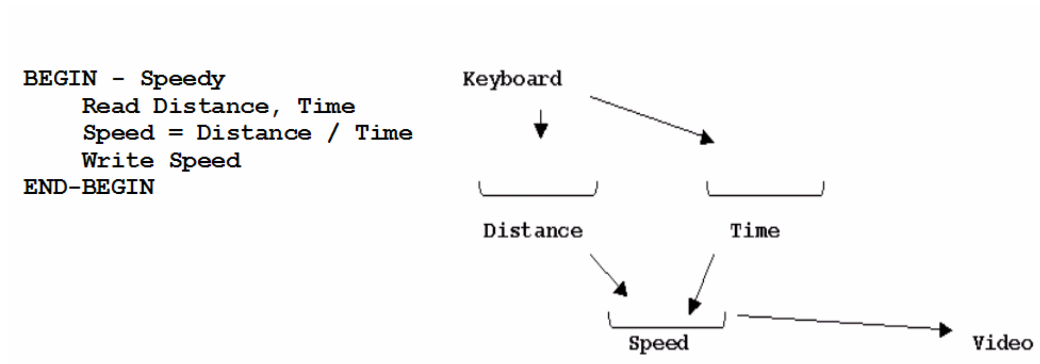

Figure 4.7. The program *Speedy*.

A very short software program should elucidate how things go. Suppose a computer operator enters the distance covered by a car and the time spent by this car in covering it. The program *Speedy* displays the velocity of the intended car. Figure 4.7 exhibits the list of the software instructions on the left side, and the data flows through the data-fields on the right side. Let us examine the algorithm step by step.

It is evident how the software program provides a new item of information – the velocity of the car – due to the sequel of statements and the set of data-fields prepared by the software programmer. Ahead of time the programmer is unaware of the numbers that the computer operator will enter, but this ignorance does not debar the programmer from planning the overall scheme of work. He is capable of organizing the items necessary for the machine to obtain the intended results. The correlated arrangements of operations and fields constitute the factual answer to the serious problems that proved to be intractable from the philosophical standpoint so far.

The algorithm *Speedy* turns out to be a straightforward case, deterministic and static. One could suspect that advanced applications deviate from the aforementioned scheme. Many problems in Artificial Intelligence such as problems in *reasoning, planning, learning* and *robotics* require an agent to operate with incomplete or uncertain information. *Genetic programming* creates *evolutionary* applications. Advanced software applications are incomparably more complex than *Speedy*, nonetheless the previous conclusions remain true in the sense that instructions and fields are the irreplaceable elements of any software construction. The methods of work and the algorithms developed in the living environment cannot be compared with the scholastic example related above. AI solutions are intricate in a superlative manner but the substantial elements of data-processing – instructions and fields – are always the same, and these elements enable us to infer further conclusions.

9. What Software Programmers Can and Cannot do

Thinkers have explored the limits of the computer tasks. For long, they have inquired what a computer can and cannot do.

In formal terms a mathematical problem is said to be '*computable'* if it can be solved in principle by a computing device. Common synonyms for '*computable*' are '*solvable*' and '*decidable'* (Cooper 2003). Hilbert believed that all mathematical problems were solvable, but in the 1930's Kurt Gödel, Alan Turing, and Alonzo Church showed that this is not the case. There is an extensive study and classification of which mathematical problems are computable and which are not. Later theorists investigated the *computational complexity classes* that classify computational problems according to their inherent difficulty (Epstein et al 1990).

Theoretical researches are still in progress. A number of intricate questions are waiting a complete answer but instead of arguing from the abstract stance I prefer to discuss practical issues on the basis of the elements just introduced.

Software designers obtain the project specifications of what a computer system is required to do for the customers and investigate whether a set of instructions and fields meets with those requirements. There is no cookbook for software designers: sometimes the algorithm is immediate; other times it is hard to devise the algorithm; lastly, a software solution may be simply unachievable in practice. Normally it is not a question of abstract 'decidability' in companies, businesses and institutions; it is a question of several pragmatic details which turn out to be fuzzy, uncertain or even unknowable by professionals. One could hold that *a computer can treat all the problems that software experts are capable of analyzing and converting into instructions and data fields*. Instructions along with fields are the 'silver bullets' for software practitioners but there is no simple and infallible rule to implement an effective software application.

2. TWO MODELS ARE NEEDED FOR A DIGITAL SYSTEM

By definition, a digital system manipulates signs and facts make clear that a computer processes *E* and *NE* using distinct units. A conversion process modifies the forms of information while meanings are kept untouched. I put forward the following formalism to depict the basic property of converters:

$$E_{IN} \neq E_{OUT}$$
$$NE_{IN} = NE_{OUT}$$

Data-processing carries on novel meanings while signifiers are uniform from the physical viewpoint:

$E_{IN} = E_{OUT}$
$NE_{IN} \neq NE_{OUT}$

Practical experience shows how definitions [1] and [2] are true in the working environment but the analysis is incomplete and one may wonder:

What is the overall model of a digital system?
How can converters and data-processors work together?

Inasmuch as digital engineers follow a pattern of rational behavior, I shall deduce the answers on the logical plane, later on we shall validate the inferences.

A. Star Model

Converters adapt the pieces of information in aid of the processor that works with physically uniform signals. Therefore a digital system should be equipped with one data-processing unit placed at the center and a number of conversion units placed all around it. The *radial model* (or *star model*) should exhibit the logical displacement of parts in a computer system.

Let us see whether computing machines comply with the inferential reasoning that I have derived from the definitions of operations [1] and [2].

Computer designers coined the term '*central unit*' for the data-processing unit because of its position, while the conversion units were called *peripherals* or *input/output units*. The latter devices lie around the former and prove that the radial configuration is correct.

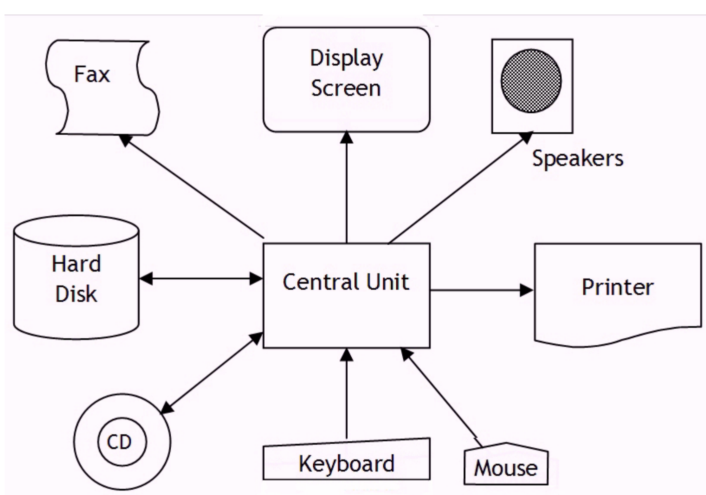

Figure 4.8. Radial Model.

The radial model identifies an external unit even if integrated and rather hidden. For instance, the star model pinpoints the hard disk as a distinct input/output unit whereas the central box in a personal computer encapsulates the disk drive and conceals this unit from human sight.

A SP digital system usually has a few peripherals, whereas a GP system offers a variety of services and is equipped with several units which operators can change at their will. Devices lately sold as regular stand-alone devices, are becoming peripherals of modern computer systems. This modern visible tendency in the computer sector is currently known as *digital convergence*. The USB (Universal Serial Bus), a plug-and-play interface between a computer and add-on devices such as audio players, joysticks, keyboards, telephones, scanners, and digital cameras, reinforces the connectivity of a portable or fixed system. The Internet facilitates digital convergence as hypermedia information can be instantly transferred from the Web into a personal computer (Sullivan et al 2004).

Probably the world in the future will become a place without books, letters, post cards, billboards, telephones, photographs, movies, televisions, stereo systems, and fax machines. In lieu of the media that we now take for granted there will be the one digital medium that will take full advantage of the star model.

B. Hierarchical Model

The central unit needs homogeneous signals and can run provided the input and output flows are homogeneous. Process [1] adapts the physical forms of data in function of process [2] and it may be said that *the peripherals are servants of the central unit.* Peripherals appear to play a subordinate role with respect to the central unit according to the present reasoning. A principle of operational mastery which may be associated with the arguments treated in (Pattee 1973) should govern the computer system from the current viewpoint and the *hierarchical tree* (Salthe 2001) should complete the description of the computer hardware.

Let us check whether the previous reasoning deduced on the basis of definitions [1] and [2] is correct.

Common practice shows how it is not sufficient for a conversion unit to be connected; this unit can only run under the central unit grant (Brookshea 2004). There is no other way to avoid conflicts amongst the peripherals; an input/output unit is idle unless the central unit empower this unit. In other words all the peripherals lie under the operational supervision of the central unit.

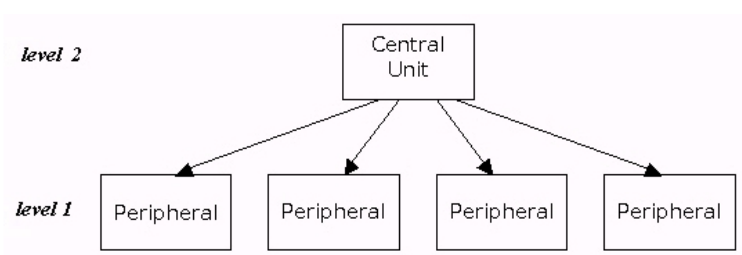

Figure 4.9. Hierarchical Model.

Computer designers strengthen this natural hierarchy and place the *control unit* (CU) – a circuit specialized for the operational control of the overall system – into the central unit. Normally CU integrates into a chip named *central processing unit* (CPU) and the acronyms CU and CPU are frequently used as synonymous (Green 1988).

A computing machine is equipped with hybrid devices, and CPU orchestrates the varied components in order to avoid conflicts amongst the parts and to optimize the overall performance. In this way the CPU reinforces the supervision role played by the central device and one can conclude that the tree model turns out to be even truer in this case.

Several hardware procedures relate minutely how CPU executes its operational supervision. I cannot look closely at those rules, and confine myself to three examples that should clarify the style of the central unit and should furnish further evidence of the correct application of the tree model.

- A peripheral device cannot receive or emit data unless the CPU enables this peripheral to work. Any external operations cannot commence without the central authorization. The periphery does not run autonomously but relies on the center.
- As soon as a peripheral ends a task, the CPU checks the results of the input/output operations that have just come to an end. In other words, all the outcomes are tested and assessed with care by the central device that undertakes the appropriate initiatives to manage the situations.
- When a special event occurs – e.g. a circuit breaks down – every running operation comes to a halt at that moment and the Central Processing Unit takes full control of the machine and handles this event. This proves that the CPU leads all the units during ordinary jobs and special occurrences alike.

CONCLUSION

This chapter proves how the digital systems share a universal structure. A logical chain leads us *from the semiotic properties of signs to the system units,* and in turn to *the radial and the hierarchical models* which depict the hardware structure of any digital system.

Facts demonstrate that systems conform to the logical conclusions developed in this chapter independently of whether a system is stand alone or connected to a net, whether a system is embedded in a machine or a big supercomputer. One can verify the star model and the tree model in cellular phones, hi-fi pods, satellite navigators, flight control equipment and so forth.

In the second half of the twentieth century we observed the astonishing proliferation of digital machines which invaded the market and replaced analog devices. After millennia, the beneficial support of the analog paradigm to human progress gave the impression of ceasing all at once. The analog appliances which boosted the advance of humankind seemed to be definitively destined to vanish despite their outstanding support to civilization. As dinosaurs suddenly became extinct million years ago, so the analog devices appear to be destined to certain death. This impression however is misleading; we have seen how analog devices – the converters – occupy prominent positions within the digital system structure. The accurate analysis just conducted leads to a rather paradoxical conclusion: *no digital computer can run without the support of analog technologies.*

A. Annotations on Current Literature

Commentators usually refer to the "*First Draft of a Report on the EDVAC*" written by von Neumann (1993) and to other earlier contributions (Booth 1960). John von Neumann presents the overall organization of a system equipped with a stored program and the '*von Neumann model'* constitutes a cornerstone in Computer Science.

Figure 4.10. Input-Process-Output Model.

Unfortunately theorists do not have a strong inclination to plunge into the general features of digital systems. Several papers and books describe the von Neumann model using a linear graph that includes three blocks (Forouzan 2003). The *Input-Process-Output* (IPO) model sometimes shows the memory, the CPU and other details; these variants are inessential for the moment. One can remark:

- The IPO model is consistent with the radial model but is linear and conceals the versatility of the general purpose computers. IPO hides the physical variety of messages held by systems. IPO simplifies the hardware structure to the extent that IPO prevents a student from understanding the *digital convergence* and its enormous factual impact. At least, it may be said that *IPO distorts the computer hardware.*
- The linear chain implies – in conformity with *Markov's chain theory* – that the central block depends on the input block; instead the contrary is true. The process unit factually governs all the external operations and an input unit cannot run until the central unit enables the operation in advance. Thus *IPO gives misleading information.*
- Surhone and others (2010) hold that IPO is an interdisciplinary tool. In fact, a lot of systems in various sectors of production – e.g. agriculture, mining, hydraulics, etc. – comply with the IPO graph in the sense that several automatic systems are served by units that introduce raw materials and bring out finished goods respectively. IPO is flexible however the labels '*input'*, '*process'* and '*output'* sound rather generic in the computer sector until a commentator specifies in what the peripheral processes and the central processes consist. If theorists do not specify the special treatments undergone by information crossing the computer system, *the meaning of the IPO model appears very approximate and the overall purpose of the system is ephemeral.*
- Scholars do not accurately examine the real origin of data-processing and are inclined to conclude that data-processing and calculation come to mean the same thing; conversely nowadays *digital systems treat various forms of data in the real world* and do not restrict their intervention to numbers.

The lack of an accurate description of the system hardware entails that technical manuals seem to be written as cookbooks. As a cookbook provides a guide for expert cookers, so modern, up-to-date textbooks are able to assist learners in the use and installation of computers. Theoretical constructs are perhaps unnecessary for those who become wizards at the operative level but prove to be essential for those who manage or improve an ICT infrastructure. The *sciolism in Informatics is myopic* and does not provide thorough knowledge. We shall see next how a number of computer disasters are caused by poor education (Dvorak 2004); in fact, the behavior of digital designers appears intelligent, but the benefits deriving from their conduct sometimes do not fully benefit the people involved due to scarce explanations. The consequences of the approximate theoretical support on computing are not of negligible significance and value.

Von Neumann adds this comment to the paper quoted above:

> "The three specific parts [that make the central unit n.d.r.] correspond to the associative neurons in human nervous system. It remains to discuss the equivalents of the sensory organs or afferent and the motor or efferent neurons. (…) These are the input and output organs of the device, and we shall now consider them briefly. (…) There remains (…) the necessity of getting the original definitory information from outside into the device, and also of getting the final information, the results, from the device into the outside." (Neumann 1993)

Von Neumann compares input/output devices to the human sensory organs and associates data-processing to the neural nets theorized by McCulloch and Pitts in those years. This analogy with the human nervous system raises vivid suggestions (Boden 2008) but appears rather questionable on the intellectual plane. Neumann aims at explaining a rather mysterious topic – say the peripherals and the data-processing unit – by using a comparison term that is even more enigmatic. Usually scientists follow the opposite direction. Doctors illustrate a biological organ using an artificial device, for instance, the bones of the arm and the forearm are compared to mechanical levers; the heart looks like a pump, the blood vessels are similar to pipes. In fact a machine, designed and built by humans lies normally under the full control of the machine's authors; by contrast living beings are ready in Nature and scientists have to decipher their complex functions. Biologists and physicians help themselves by means of technical concepts which make it easier to understand biological parts. Neumann seems to adopt the opposite method which sounds rather strange; he attempts to elucidate the functions of a computing system by means of the parallel with the human brain which is still a challenge for researchers world-wide.

All this should prompt accurate studies of the structure of the digital systems. The basic properties of the computer hardware should be brought into the open and its merits debated, but this project has not progressed far. Theorists have not proceeded to investigate the structure of the computing machine and this passive orientation is not a trifling matter, since a number of fallacies and misconceptions, which are caused by the negligent behavior of theorists, may be found in the current literature. The magnitude of the negative effects in the living environment appears even more manifest.

BIBLIOGRAPHY

Boden M. (2008) - *Mind as Machine: A History of Cognitive Science* - Oxford University Press.

Brookshea Glenn J. (2004) - *Computer Science: An Overview* - Addison Wesley.

Booth A.D. (1960) - The Future of Automatic Digital Computers - *Communications of the ACM*, 3(6), pp. 339-341.

Cooper S.B. (2003) - *Computability Theory* - CRC Press.

Doyle L.F. (1999) - *Computer Peripherals* - Prentice Hall.

Dvorak J.C. (2004) - *The Bottom 10: Worst Software Disasters* - PC World Magazine, 8.

Epstein R.L., Carnielli W.A. (1990) - *Computability: Computable Functions, Logic and the Foundations of Mathematics* - CRC Press.

Forouzan B.A. (2003) - *Foundations of Computer Science, From Data Manipulation to Theory of Computation* - Thomson.

Green D. (1988) - *Digital Electronic Technology* - John Wiley & Sons Inc.

Koch C., Tononi G. (2008) - Can Machines Be Conscious? - *IEEE Spectrum*, 45(6), pp. 55-59.

Kraft G.D., Toy W.N. (1979) - *Mini/microcomputer Hardware Design* - Prentice-Hall.

Neumann von J. (1993) - First Draft of a Report on the EDVAC - *IEEE Annals of the History of Computing*, 15(4), pp. 27-75.

Norton P., Clark S. (2002) - *Peter Norton's New Inside the PC* - Sams Publishing.

O' Sullivan D., Igoe T. (2004) - *Physical Computing* - Muska & Lipman.

Pattee H.H. (1973) - *The Organization of Complex Systems - in Hierarchy Theory: the Challenge of Complex Systems*, Pattee H.H. (ed), Braziller, pp.1-27.

Salthe S. (2001) - *Summary of the Principles of Hierarchy Theory* - MIT Press.

Scott M.L. (2008) - *Programming Language Pragmatics* - Morgan Kaufmann.

Surhone L.M., Tennoe M.T., Henssonow S.F. (2010) - *IPO Model* - Betascript Publishing Co.

Stojanovic D. (2009) - *Context-Aware Mobile and Ubiquitous Computing for Enhanced Usability* - IGI Global Publ.