University of Lagos Press Tel: 825048

COMPUTER VISIONS OF REALITY

By

Olayide Abass

UNIVERSITY OF LAGOS PRESS - 1997
INAUGURAL LECTURE SERIES

© Olayide Abass

First Published 1997

ISBN: 978-017-324-2

by
University of Lagos Press
P.O. Box 132, Unilag Post Office
University of Lagos
Lagos, Nigeria

COMPUTER VISIONS OF REALITY

INTRODUCTION

This being the foundation Inaugural Lecture in Computer Science, it will be appropriate to devote part of the lecture to explain what the discipline is all about and what makes it a science. The latter part of the lecture will be devoted to the theme of this lecture *Computer Visions of Reality*. We will show how computers can generate scenarios and therefore visions of real life systems that define our world.

1.0 SCIENTIFIC PROCESS

Ordinarily, we would have proposed a definition of science and examine whether Computer Science satisfies the requirement of this definition. Such a definition may not help because sciences are not defined, they are recognized. Science is dynamic, as it evolves,its meaning changes. This evolution is not to be stopped by act of defining. To illustrate the futility of definition, UN Agency, a few years ago, set up a committee to define science, to distinguish between (1) a scientist and a technologist (2) a technologist and a technician. The committee spent two years without agreement on any of these issues.

For our purpose, it is sufficient according to Ackoff (1968) to recognize science as a body of knowledge generated through a *process*. We are interested in the *process* that generates the knowledge and not the knowledge itself.

It is easy to have a common understanding of the four essential characteristics of the scientific process

(1) Science is considered as a process of inquiry for
(a) Answering questions
(b) Solving problems
(c) Developing more effective procedures for answering questions and solving problems.

It should be noted that not all inquiries are scientific. There are common sense inquiries.In most cases distinctions between scientific and common sense inquiries are

(i) Common sense inquiries deal more with immediate problems and practical problems;
(ii) Scientific inquiries are more quantitative, while common sense inquiries are qualitative;

For example, an American Philosopher, Dewey (1938) observed " The problem of the relation of the domain of common sense to that of science has notoriously taken form of opposition of the qualitative to non-qualitative, largely, but not exclusively the quantitative"

These two distinctions may not always exist for in applied science we deal with problems that are immediate and practical. The distinction in terms of one being the more quantitative than the other also breaks down, for as Dingle (1953) pointed out, one outstanding scientific achievement of the nineteenth century was the theory of evolution which has nothing to do with measurement, it is concerned with qualitative changes and treated as such.

This shows that there is considerable overlap between problems and questions investigated scientifically and non - scientifically. Hence we need to look for other characteristics:

(2) Science is also characterized by goals of self-perpetuation and self improvement. It is a common institution that pursues an ideal: increase

without limit our knowledge and ability to answer questions and solve problems. This implies the requirement in scientific research that it must be conducted in such away as to increase the efficiency of future research.

(3) Scientific inquiry is controlled. A process is controlled to the extent that it is effectively directed towards attainment of desired objectives.

(4) In scientific inquiry, experimentation is necessary. Through experimentation, factors, temperature, humidity, pressure etc, are controlled interactions between factors are studied, and measurements are made. These days, it must be pointed out that experimentation can be carried out without physical manipulation. Experiments can be performed on the computer through simulation. Since pseudo-random numbers are generated and used during this process, experiments are repeated/replicated under desired conditions. This is the process that leads to visions of reality.

2.0 DOMAIN OF COMPUTER SCIENCE

One of our objectives in this lecture is to contribute to the discourse as to whether computing is a science or an art. It is necessary therefore to ask and answer the question, "what do we do in Computer Science?"

An encyclopedia of Computer Science has offered the following definition:

" Computer Science is concerned with information process, with information structures and procedures that enter with representations of such processes and with their implementation in information processing systems. It is also concerned with relationships between

processes and classes of tasks that give rise to them."

This definition is not too satisfactory because there is the need to distinguish between "information" and "data" even though the two words are often used interchangeably within the discipline of computer science. Information results from processing data. In order words, information is endowed data. The endowment results from processing. We therefore propose that Computer Science is the study of data in the following sense:

(1) Theoretical foundations which describe what sort of data can be produced from raw data;
(2) Machine that holds the data;
(3) Languages for communicating instructions to the machine;
(4) Structures and methods for organizing data in the machine; and
(5) How to maximally utilise the resources of the machine.

The term "data" has been used here in the generic sense. This data, can be numeric, alphabetic or alpha-numeric. It can be propositions, strings, symbols,text, etc.

2.1 Theoretical Foundations

The theoretical foundations of abstract computer science deals with abstract machine, algorithms, recursive functions and computations.

(a) Turing Machines

Minsky (1967) remarked:
"Man has within a single generation found himself sharing the world with strange new species- computers and computer-like machines, though not all of us deal directly with computers, we are all falling under their ever growing sphere of influence, and thus we all need to understand their capabilities and limitations".

Thirty years ago when Minsky expressed this view, the modern computer was just a toddler. The sphere of its influence has expanded several thousands of times over what it was then. Computers, then were some kind of back boxes which only specially trained professionals could touch. Now the situation is much different; many people, not necessarily professionals can now operate the computer. What is still true is that many of us do not appreciate its capabilities as a tool for problem solving.

Over a decade before modern stored-program computers were built, Alan M. Turing (1936), an English Mathematician working at Cambridge University proposed a class of abstract machines called Turing Machines (TM). A TM can be considered as a computer device. These machines are of utmost simplicity. Turing proved, using mathematical logic, that any *computable function*, no matter how complex, can be computed by a machine in this class of Turing machines, if the machine can be provided with as much time as possible and infinite auxiliary storage. This process is known as " Turing's thesis".

The contribution of Turing is held with such a high regard by the

computer scientist that an annual Turing lecture has been established by Association Of Computing Machinery ACM. Universal Turing Machine (UTM) can be regarded as the theoretical prototype of a real general-purpose computer.

It is perhaps necessary to explain that computation here, is not restricted to arithmetic manipulation: Computation, here will include string manipulation, parsing, language translation etc. This shows that class of processed data that can be obtained from raw data is very wide. Other early, notable contributions to theoretical computing devices include E. Post (1936) Markov (1936) and Church (1936).

Post & Markov laid the foundation for Chomsky's theory of formal languages. Their work also, has found use in string processing and pattern matching languages such as SNOBOL, PLANNER. Although the approaches followed by Turing (idealized element) and Post and Markov's (formal symbol manipulation) were based on methods that initially appeared to be different from each other they later turned out to be equivalent from point of view of understanding theory of algorithms. Besides Turing Machine and those of Post and Markov there are many others. (Krishnamurthy ,1991)

An abstract computing machine must be capable of
(i) reading and interpreting input and intermediate data
(ii) storing data
(iii) carrying out primitive mechanical operations as required in the algorithm.
(iv) mechanically controlling the sequence of steps

(v) Outputting the report.

From the above it is clear that Turning and others have settled the issue of whether not a machine can be instructed to solve problems and answer questions. There remains the problem of how to practically instruct the machine. Indeed Von Neumann (1948) in a seminary paper had proposed that the data and program be simultaneously be input into the machine. This gave birth to the concept of stored-program computer. The program is the set of instructions the computer must fellow to process the data. A program is a coded algorithm. The relationship between the program and data is analogous to that between recipe and ingredients for preparing a meal. For a cook to prepare a dish, the ingredients (data) and the recipe (algorithm) must be supplied.

(b) Algorithm:

Before we define what the term algorithm means in computer Science and what are the properties of an algorithm, it is useful to trace the origin of the word " algorithm".

The word "algorithm" is quite interesting; at first glance it might look as if one intended to write *logarithm* but permitted the first four *letters*. If one looked at Webster's *New World Dictionary* before 1957, one would not find it;Instead one would find the old form "algorism with its ancient meaning, the process of doing arithmetic using Arabic numerals. Following the middle ages, the origin of the word was in doubt and early linguists attempted to guess at its derivation by making combinations like algoris (painful) to arithmos (number), others thought the word comes from King Algor of Castile. Finally, historians of Mathematics found the true origin of the word. It comes from the name of a famous mathematician, Muhammed Ibu Musa ab Abdullah al Khorezimi al Madjusi al Qutrubulli who was born about 810 A.D. in Khorezm (now Urgench in Uzebekistan which was part of USSR. He served as astronomer to the Caliph Abd-Allah al mamum (The Trust worthy) who inherited the Bayt al-Hikmah (House of Wisdom or Academy of Sciences) at Baghdad from his father Harun ar-Rashid (The Righteous) and brought to its peak. Al-Khorezimi's known works are ten in number and include:

(I) Kitab az-zil al-sindhine, a set of astronomical tables including the first sine and cotangent tables,

(ii) Kitab hisab al-adad al-handi, an arithmetic text.

(iii) Kitab al- Muhtasar fi hisab al-gabr wal-muqanbalah, a compact text on solution of equations.

This third book latinized as "Ludus algebrae et al-mucgrabalaeque gives the world algebra.

The second book, latinized as "Algorithm de numero Indorum", brought Hindu number system to Arab world and the resulting Hindu-Arabic number systems to Europe, that is, the system of "Hundreds" "Tens" and Units.

In medieval times arithmetic was identified with his name rendered as 'Algorismus' The formula dixit Algorizmi (thus spoke Al-Khorezimi) was a hall mark of clarity and authority. Has text in arithmetic included all basic arithmetical processes by which numbers could be added, multiplied, subtracted, divided, doubled, halved and their square roots extracted. It had a chapter on business calculations. Today has name; "Algorithm" survives as logical sequence of steps for solving a given problem. It is a concept at the heart of practical and theoretical Computer Science and the mathematics of computability.

It must be noted here that modern computers have a lot of applications in business. An algorithm for solving a problem must exhibit the following properties, Knuth(1968)

Finiteness: An algorithm terminates after a finite number of steps, i.e when the algorithm is mechanically executed, it comes to a stop after executing a finite number of assignment, decision and repetitive steps.

Definiteness: Each step of the algorithm is precisely defined, i.e actions to be carried out are

rigorously and unambiguously specified and the sequence of steps to be executed are unambiguously specified.

Generality: The rules of the algorithm are complete so that it can solve problems of a particular type (for any data) for which it is designed.

Effectiveness: All the operations used in the algorithm are basic and are capable of being performed mechanically.

Input-Output: An algorithm has certain precise inputs or initial data and outputs are generated in the intermediate as well as the final step of the algorithms.

One important observation to be noted here is that the properties of algorithm stated above are problem independent. We do not require different set of properties for algorithms for solving numerical problem (finding roots of polynomials) or solving non-linear equations and another set of properties for an algorithm for sorting a set of data or translating one text.

This makes for possibility of using computers to solve different types of problems, and hence accounts for the ubitiquity of the computer. By varying our algorithm, we can take on and solve different types of

problems. This is why its influence has been pervasive. It is applied in Engineering, Law, Transportation, Economics, Educations, Music, Architecture, Banking, Insurance, Medicine, Printing, etc. Algorithm development is like poem writing; whose limit is determined only by capacity and ability to imagine. Since the capacity of man to think and imagine is unlimited, areas of application of computer will continue to grow.

(c) Complexity of Computation

The subject of complexity of computations deals with quantitative aspects of computational problems. Given, a computational problem there are several possible algorithms for solving the problem. By a computational problem we mean a problem such as evolving an algebraic expression, sorting a file or parsing a string of symbols. With each algorithm, there are associated cost functions like, the number of computational steps as a function of the size of the problem, memory space requirements, for the computation, and in hardware implemented algorithms, circuit size and depth.

Given a problem P, which is computable, the following questions arise:

(a) What are good algorithms for solution of P.?

(b) Can one establish a lower bound for one of the cost functions associated with the algorithm?

(c) Is the problem intractable in the sense that no algorithm will solve it in practical feasible time?

These and similar questions have engaged the minds of computer scientists for sometime.

3.0 LANGUAGES FOR COMMUNICATING INSTRUCTIONS TO THE COMPUTER:

Since most of the existing common computers in use world - wide today are von Neumann machines, that is stored-program computers, the instructions (algorithms) they will use to process data/ solve problems must be stored in the computers along with the data. A computer program is a coded algorithm, coded in a language which the computer can understand. Even though the situations may change in the nearest distant future, when we may be able to communicate with computers by voice, presently we do so through programming languages.

Since 1950's there has been four generations of programming languages:

1st Generation:		Machine Languages (Low level programming languages)
2nd	,, :	Assembly Languages (Low level programming languages)
3rd	,, :	Procedure Oriented Programming Languages
4th	,, :	Problem Oriented Languages
5th	" :	Object Oriented Languages

These programming languages make development of other programs easy or use of computers by specialists easy. The developments in Computer Science have always been into two dimensions.

---- Dimension of Hardware

---- Dimension of software. (Programs, programming languages, algorithms,structures of data, operating systems, compilers,etc)

We shall deal with the distinction between the hardware and software later. It is sufficient to state here that a computer will be another piece of electronic equipment without the software. It will be an incomplete machine, like a piece of musical instrument waiting for a skillful player to play it. Since a software is a collection of programs, it is essential to spend a few more minutes on programming languages development in Computers Science.

As stated earlier, there have been four generations of programming languages.

3.1 Machine Language

Programming of earlier computers was achieved in a very cumbersome fashion as the machine needed to be rewired in order to run a new set of instructions. The focus was to get the work done rather than with the conceptual difficulties of problem solving. A typical program to add two figures would look like:

```
0   00000010101111001010
1   00000010111111001000
2   00000011101010110000
3   00000010101111100110
4   00000101111111000100
5   00000011001111001000
6   00000011001110101000
```

The first ten binary digits correspond to the *address* of some single item of data. This datum is either to be used by the computer or alternatively the result of some calculation is *stored* at this address. The second set of ten binary digits indicate the operation to be performed, an addition, subtraction etc. This series of digits constitute a program which the computer can interpret directly and is therefore called a *machine language* program. One of the problems associated with programming in machine language is that such programs cannot be easily written without human errors. Secondly, such programs are un-intelligible to human beings, Thirdly, they are very cumbersome.

3.2 Assembly Language:

To eliminate the errors associated with machine language, symbolic languages were developed and the first of such languages is Assembly language. The above machine language written in Assembly language will look like

0. 10 L, Load 'A' from address 10
1 11 A, Add to 'A' the contents of 11
2 12 S, Store this value in 12
3 10 R, Reload 'A'
4 11 X, Multiply by contents of 11
5 12 A. Add to the value of 12
6 12 S, Store this value at 12

If X is stored in 10 and value Y in 11, then this program is calculating.

$(X + Y) + (X * Y)$

Whose result will be stored at address 12. The program will then be converted into machine language by a program called assembler. This program is more readable and understandable to human beings. However, it suffers from one great drawback. The assemble language is not unique in the sense that different computers have different assembly languages. Thus, it is not possible to transfer one assembly language program for one type of computer to another. Furthermore, assembly language fails to reflect the manner in which we conceptualize problems. For this reason, efforts have been focused on development of high level languages.

3.3 High Level Programming Languages:

One of the greatest challenges Man has ever faced was that of developing a language that both human beings and a machine can understand. A high level programming language is a language which is close to say, English language, and can be understood by a machine (computer).

Before the invention of computer this problems/questions never existed!!! The natural way a man learns a language is to pick it from his immediate environment, determined largely by his parents, during childhood. This language is usually defined as his mother-tongue. He then learns other languages through formal education or his living in a different environment where a different language is spoken and forced to learn that language by survival instinct.

Before we mention examples of high level languages, it is important to briefly point out the contribution of Chomsky. Noam Avram Chomsky was a son of a Hebrew scholar. He was born in Philadelphia and studied mathematics, philosophy and linguistics. He had been interested in politics and contemplated dropping out of college to live on a Kibbutz and work for Arab-Israeli Corporation. His work on generative grammars, which grew out of his interest in logic laid the foundations for theory of formal languages.

One of the first high level programming languages to be developed FORTRAN, FORmula TRANslation first evolved between 1954-1956 for use on the IBM 704 computer and has been backed by IBM ever since. It sustained a number of modifications before being standardized as FORTRAN IV in 1966. There has been improved versions FORTRAN 77 and FORTRAN 88. Today, there are over 500 programming languages. Some of the more popular ones currently are COBOL, SNOBOL, Pascal, BASIC, SIMULA, PL1,APL,LISP, PROLOG, C, ALTRAN, WATFOR, SIMPLAN, SIMSCRIPT, GPSS, DYNAMO, GASP, and Ada.

There are very many reasons for development of these languages. FORTRAN was developed for solving scientific problems, while COBOL, COmmon Business Oriented Language: was developed for handling business problems, Pascal was developed in the late 1960s by Niklaus Wirth primarily as a teaching language and as such was used within departments of Computer Science. However, the language is now in use outside of Computer Science departments because it has a structure which makes it very ideal for structured programming.

Ada, was developed from the sponsorship of American Department of Defense with the aim of developing a language that could support parallel processing besides sequential processing.

The language is named after Ada Augusta, Countess of Lovelace (1815-52) who was the only child of Lord and Lady Lovelace. She studied mathematics under Mary Somerville. She worked with Charles Babbage(1791-1871), an English mathematician in Cambridge University. He held the Lucascan Chair of mathematics at Cambridge. Babbage's Analytical engine, though not completed is universally accepted as the world's first computer. Ada is regarded as the first computer programmer.

Developing programs using third generations languages can be very expensive, also they take long periods of time. This is why computer software used to cost so much, even more than the hardware. A third generation language, must have a compiler, which translates the language to machine language.

3.4 Fourth Generation Languages (4GLs):

One of the positive effects that resulted from the software failures that occurred in the early 1980s was the emergence of a new branch of study in Computer science, Software Engineering, whose primary purpose is to develop tools for making the development of application programs in a much easier and less costly way. This became necessary as consumers were denied benefits of falling prices of

computer hardware because of high cost of software. This cumulated in the intensification of efforts to develop 4GLs. These 4GLs are problem oriented languages.

(a) DBMS: Dbase IV, Access, Oracle, Rbase, Paradox, etc. This set of 4GLs, are useful for developing databases.

(b) CASE: (Computer Associated Software Engineering .) This is used for developing Business Application programs (Inventory, Payable, Account Recovered, Payroll etc.)

(c) CAD: Computer Assisted Design, for use in Engineering and Architectural designs.

(d) Word Processors (Word Perfect, MS World, Ventura etc.) facilitate office automation, and desk top publishing through Macros Script Languages.

(e) Statistical Packages: (SPSS, BMDP, NCSS, etc.) These packages make statistical data analysis easy and cost effective.

(f) Simulators : (GPSS, SIMPLAN, GASP etc.) This category makes simulations easy to accomplish.

3.5 Fifth Generation Languages(5GLs):

Whereas the 3rd generation languages are Procedure-oriented and the 4GLs are Problem-oriented, 5th generation languages combine the features of both; in that they are Object-oriented. Objects as defined by their encapsulation of procedures and data (problems). Examples includes:

(a) SmallTalk, C++, Java

(b) Access, VisualBasic, Delphi

The arrival of these 5GLs, and availability of 5th Generation Operating Systems have made the use of computers by non programers more friendly. Indeed, without these 5GLs, the present wide use of computers will not have been possible, and consequently we would not have been able to take advantage of capabilities (very fast speed, enormous memories,etc) of present day computers.

4.0 OPERATING SYSTEMS: HOW TO MAXIMALLY UTILIZE THE RESOURCES OF THE MACHINE.

One piece of software that makes the computer user friendly is the operating system. Before the first generation operating system was developed, computer operation was manual; each job step required a manual intervention. Computer had to be programmed in either machine language or assembler, and it was not possible to provide multiple application support. These restrictions coupled with high cost of computer then limited use of computers to large business organizations, government establishments, Universities and Research Institutions as these organizations were the only bodies that could afford huge financial layouts required.

Operating Systems often shape users' views of the computer more than the hardware itself. Computer manufacturers these days must build hardware to run on the platform of an existing operating system. Windows 95 is an example of such operating system.

An Operating System is a resource manager and the primary resource it manages is the computer hardware in the sense of

(i) defining the "user interface"

(ii) sharing hardware resource among users
(iii) allowing users to share data among themselves
(iv) scheduling resources among users
(v) facilitating input/output
(vi) recovering from errors

The key resources an operating system manages are

(a) Processors
(b) Storage (memory)
(c) Input/Output device

It interfaces with

(a) Computer Operators
(b) Applications programmers
(c) Systems Programmers
(d) Administrative Personnel
(e) Programs
(f) Hardware

Since the early 1950s, when the first operating system was developed, there have been very many operating systems. In fact every computer manufacturer needed to develop his own operating system for each model of computer it manufactured. Things have changed tremendously now. One can specify the type of hardware one wants and the type of operating system wanted with it. The two need not come from the same manufacturer. The evolution of operating system design and implementation has paralled the evolution of programming languages. So we can talk about generations of operating systems.

4.1 1st Generation (Late 1950s, and early 1960s).

The General Motors Research Laboratories are credited with implementing the first operating system on their IBM 701 compiler. In 1955, General Motors and North American Aviation is cooperated in the production of an OS for their IBM 704 computer. These OSs could support programme writers in FORTRAN and COBOL, and amongst other things, provide for primitive data services with error recovery.

4.2 2nd Generation (Early 1960s)

Examples of second generation of operating systems are OS/360 on IBM systems/360, and MCP for Bearroughs 5000. They could support high programming languages like FORTRAN, COBOL, ALGOL, APL, BASIC PL/1. They support the following operations: multiprogramming, primitive work-load management, primitive tuning (device, core allocation spooling), remote job entry, and multiprocessing. In addition they are device independent in data accessing, and provide primitive software error recovery, and time-sharing processing.

4.3 3rd generation (Mid 1960s - 1970s).

The third generation operation systems like MVS, OS/VS on IBM system/370 and were multi code systems. Some of them could simultaneously support batch processing, real-time processing and multiprocessing. They were very large and very expensive, like them and took much time to develop. A notable exception is this is the UNIX system developed at Bell Laboratories.

4.4 4th and 5th Generations (middle 80s - 90s)

Majority of the OS in the generation are for personal computers, OS, MS DOS, Windows, Windows NT, Novell etc.

5.0 EVOLUTION OF COMPUTERS (Machines that hold the data)

The need to expand the mental capability of man has long been felt, dating back to over 3,000 years. It began by carving motifs into pieces of wood or bones. These tally sticks as they are now called became universally applied and were finally superseded by use of paper and ink, a process which started in China. Instead of tally sticks, elaborate method of tying knots on ropes was used in the Inca empire in South America. About 3000 years ago, abacus was introduced in China as a tool for doing arithmetic. In 1630, the slide rule was introduced in Europe and became a popular tool for the engineer, while John Napier's work on logarithm tables was published in 1614.

5.1 Early Machines

As Commerce became relatively more "sophisticated" in the 17th and 18th centuries, the ancient calculating tools became inadequate and consequently attempt began to build calculating machines.

Amongst early attempts were those of the mathematicians Pascal (1623-62) and Lebniz (1646 -1716). Most of the early attempts at

automation were centered on the weaving industry Basie Boughon is credited with inventing in 1725, the idea of using a perforated tape to control the production of ornament patterns. His ideas were eventually left to Jacquard to put into practice, and by so doing produced a really successful widely used draw loom.

Charles Babbage (1791-1871) son of a wealthy banker, and founder of the Analytical Society in Cambridge University and holder of Lucasian Chair of mathematics in the University proposed building a Difference Engine. He received financial backing form the British Government in 1823 He was given seventeen thousand pounds. Even with this backing, he could not complete this project and the British Government withdrew her support in 1842. Some years later a Swedish engineer George Schuetz successfully completed a difference engine based on Babbage's idea a copy of which was bought by British Government in 1864.

What Babbage really got out of his attempt to build a difference engine was the motivation we had towards the concept building a general purpose automatic calculating machine, the analytical engine. But like is predecessor, the analytical engine was never completed.

Seventy years later after Babbage's death, his dream came true when a German, Konrad Zuse built what is believed to be the first general - purpose program controlled electro- mechanical computer. However, his effort was an isolated one as it did have very little effect on the evolution of modern computers.

ENIAC, Electronic Numerical-Integrator Automatic Calculator was built by a team drawn from Moore School and the Ballistic Research Laboratory, between 2nd April 1943 and fall of 1945. It was commissioned on the 1st February 1946.

ENIAC weighed 20 tons, occupied a space of *3000* cubic feet, contained 18,000 vaccum tubes and consummed 200 kilowalts of electric power, could add two numbers in *3 milliseconds* and multiply them in *2, 800 milliseconds*.

One mathematical tool that aided the development of other computers was Boolean Algebra which dates back to 1845 and deals with systems containing two states availability. George Boole's father was a cobbler with an interest in the construction of optical instruments. He was unsuccessful in his trade and his son George had to earn his living as a teacher from the age of 16. In 1834, Mechanic's Institute was founded in Lincoln and George's father was appointed curator of its reading room which received publications of the Royal Society and also held Newton's Principia and other mathematical classics. The young Boole spent much of his spare time in the reading room, and when the institute was given the burst of Newton, he was asked to give an address on Newton, this was his first scientific paper. He continued to study mathematics while running his own school and was awarded a gold medal in 1844 for his work on *Operator on Analysis*. In 1849, he was appointed to the chair of mathematics at Queen's College although he had no University degree. His works marked the beginning of modern logic. He died in 1864 as a result of lecturing in wet clothes.

Claude Shannon (1938) demonstrated that Boole's algebra could be applied to switching and relay circuits, thereby transforming the problems of design from electronics to algebra. Extensive work was done on this algebra and we can now deal with most aspects of computer architecture and operation without ever refering to questions of voltage, resistance and capacitance etc.
For this reason, a computer is better understood by considering it to be a logical, rather than an electronic machine.

5.2 Advances in Technology:

In some sense one can say computers have changed very little since the 1950s, in that structure as represented by its functional components have remained the same, however the materials/technology for making the components have been changing. This has had the multiple effects of reducing their sizes, reducing cost, increasing speed, reliability and maintainability.

While the first computers occupied a space of 3,000 cubic feet, we now have desktop computers, lap top computers and even palm top computers whose speeds and memory sizes are millions of times faster and bigger.

(a) 1st Generation (1945-54)

Mainframe components were made with vacuum tubes. These computers, mainframes, were large, heavy and occupied large spaces. Because the technology was vaccum-tube, they consumed much

energy and consequently generated much heat, hence requiring excessive cooling.

(b) 2nd Generation (1954-65)

Vacuum tubes were replaced by transistors, this lead to reduced sizes. The computers remained to be large mainframes.

(c) 3rd Generation (1965-)

The technology responsible for the computers in this generation is Integrated Circuits(IC's). Mini computers and micro computers became available.

(d) 4th Generation (mid 1970s - early 1980s)

Large scale Integration (LSI) became possible. This has led to manufacturing of very large scale very powerful micro computers and micro computers

(e) 5th Generation (late 1980s - 1990s)

Very Large Scale Integration is the driving technology of computers. The development of 5th generation computers was the initiative of the Japanese. The 5th generation computers will not be von Neuman sequential computer i.e. stored-program computers. It will have Artificial Intelligence and hence could be capable of voice input in human languages. Large Scale commercial production of 5th generation computers is still being delayed because of problems encountered in developing appropriate software. The remaining aspect of this lecture will be devoted to how experiments are performed in

computer science. Results of analysis of experiment give us different visions of the real- life systems under study.

6.0 THE STRUCTURE AND ORGANIZATION OF DATA IN THE MACHINE:

The structure and organisation of data can be discussed from three or four levels of abstraction. At the lower level which some refer to as the machine level, data can be viewed as a stream/sequence of binary coded strings which are organized to reflect the architecture of the machine. The codes determine the meaning to be associated with the string. Also the codes distinguish between data signals/strings and control strings.

At the lower level of abstraction which is also refered to as the the Physical level, data can be received as buckets and buffers. The emphasis at the level is the unit of communication between the processor and the physical storage medium e.g. disk, tape. In order words data can be viewed as physical buckets which are transferred from second level memory to the main memory. The structure of data is determined by the blocking factor as well as the size of the memory buffer or cliche. At the high level data is structured into records, files and databases (logical).

6.1 Database

A database is a collection of stored operational data used by the application systems of some particular organization.

The term enterprise is used as a generic term for any reasonably large-scale commercial, technical, scientific, educational, e.t.c. organization e.g. Bank, Hospital,University, Government

Dept, Any enterprise must neccessarily maintain a lot of data about its operation.

This is its operational data.The operational data for the enterprises listed above could include account data, patient data, student data, and planning data respectively.

The entities which the data have been collected can be related in some way. These relations link the entities together.

6.2 Why Databases?

In the early days of using computer for information processing, storage and retrieval of data, in enterprises, each application had its own set of data. Thus an account payable systems would be a different file from that of an ordering system. Each application would also have its own private file; often its own private tapes, disk packs, e.t.c.

In this process, data are more dispersed widely and there is little or no attempt to control it. Yet, it is often said that, next to people in any organisation, its data are the next most important. If a database is created for an enterprise it is therefore easy to identify one person, called Database Administrator whose responsibilty will be to look after the data of the enterprise. Advantages in database are:

(i) Redundancy reduction in a system where each application has its ownprivate data files there shall be redundancies in the data stored. If for example,you have a personnel application and a payroll application the data files of payroll application will be almost the same as those of personnel application.

(ii) Problems of data inconsistency in stored data can to some extent be avoided. If for instance, a fact (a student in dept. C) is represented by two entities in the database, then at some time the two entities will

not agree. If one is updated and the other is not then the database is inconsistent.
Other advantages are:
(iii) Standard can be enforced,
(iv) Security restrictions can be enforced and
(v) Data intergrity can be maintained and conflicting requirements can be belanced.

6.3 Data Independence

As stated earlier, at the dawn of application of computer for information systems,each application has its own private file in an auxiliary back storage. Thus, the type of storage, the way the data is organised (tape will be suitable for sequential data organization) will be dictated by the type of application. Database is data independent. The data base allows several applications to be run on the same data. This is the main reason why databases are developed.

6.4 Database Architecture.

The commonest database Architecture these days is relational. Others are network and hierarchical.

6.5 Database Management System

In those days databases were developed using third generation languages. This used to make their developments very slow and expensive. There are now 4th generation languages for use. Their availability makes database development very easy. Some of these 4GLs are Dbase IV, Access, Oracle, Ingress, Paradox.

Abass and Longe (1988) have developed a database for a poultry farm in Agege. This project was financed by UNESCO. The main reasons for developing the database are:
(a) To illustrate that computer applications is not limited to commercial banking, scientific or technological enterprises. Computer is a tool that can be used by anybody who cares to use it and know how to use it to improve his productivity, effectiveness and efficiency.
(b) If computer is proper utilized in the productive sector of our economy, it cannot assist us in frog-leaping.

7.0 HOW EXPERIMENTS ARE PERFORMED IN COMPUTER SCIENCE.

Traditionally, in the old Science disciplines like Physics, Chemistry and Biology, experiments are performed in the laboratory. This is still done in the hardware related aspect of Computer Science. However, when dealing with real-life systems, experimentation could only be carried out by manipulating a model of the real-life system under study. We may ask, how do we experiment with the Nigerian Educational system?

It is therefore necessary for us to consider what we mean by a model. The word model is often used as a noun, adjective or verb and in each instance it has a different slight connotation. As a noun, "model" is a representation, in the sense an architect constructs a small-scale mode of a house. As an adjective, "model" implies a degree of perfection or idealization as in reference to a model home or model student. As a verb, "to model" means to demonstrate, to reveal. Scientific models have all these connotations. They are

representations of states, objects or events. They are idealized, in the sense that they are less complicated than reality and hence are easier to use for research purposes and to manipulate, scientific models are used to accumulate and relate knowledge we have about different aspects of reality. They are used to reveal reality and serve as instruments for explaining the present, and for predicting and controlling the future.

These are physical models and mathematical models. In this lecture we shall concentrate mainly on the mathematical model.

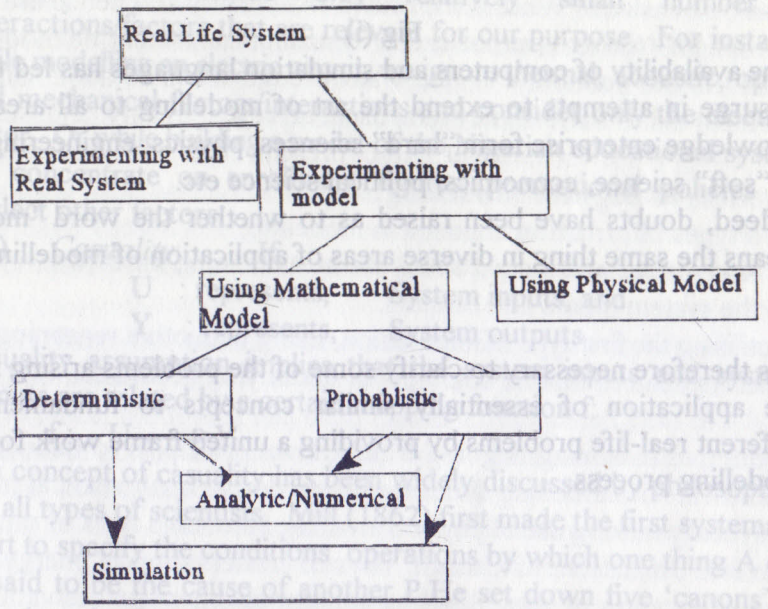

Figure 1

Fig (i)

The availability of computers and simulation languages has led to an upsurge in attempts to extend the art of modelling to all areas of knowledge enterprise form "hard" sciences; physics, engineering etc to "soft" science, economics, political science etc.

Indeed, doubts have been raised as to whether the word 'model' means the same thing in diverse areas of application of modelling.

It is therefore necessary to clarify some of the problems arising from the application of essentially similar concepts to fundamentally different real-life problems by providing a united frame work for the modelling process.

7.1 Assumptions Inherent in Modelling:

Without knowing, modellers in different areas make some assumptions. In our works in this area, we have shown that these assumptions can be summarized as:

(a) *Separability:* To a certain degree, most phenomena in the universe are interrelated. In modelling a real-life system, we confine ourselves to the interactions within the boundary of the system under study, and ignore all others. When building a model of electrical consumption in Nigeria, we neglect demographic factors.

(b) *Selectivity:* It is assumed also that the boundary defining the system shall include only relatively small number of interactions/factors that are relevant for our purpose. For instance, while modelling an electric circuit, we ignore thermal, acoustic, optical and mechanical factors/interactions and consider only the electrical factor. Or while building a model of the Nigerian educational system, we concentrate on enrollment figures, promotional policies and neglect other factors.

(c) *Casuality*: If,

U : represents, System inputs, and

Y : represents, System outputs

Casuality assumption implies that the system inputs and systems outputs are related by a certain mapping function f:

f : U------>Y

The concept of casuality has been widely discussed by philosophers and all types of scientists. Mill (1862) first made the first systematic effort to specify the conditions operations by which one thing A can be said to be the cause of another P.He set down five 'canons' of induction, the first three were concerned with one thing as the cause of another. The problem of Casuality however, continues to be a challenging and perplexing one. Bohm (1957), Born (1949), Frank (1949).

(d)*Measurability*:

It is further assumed that the input U and the output Y are measureable in numerical terms. These measures determine the state of the system. This assumption causes problems when modelling in Social Sciences. For example concepts like 'utility' in Economics are difficult to measure. This is one of the reasons why modelling is very

difficult in Social Sciences. For example, one Operations research expert was asked: 'What is the utility of a person ?' After sometime, he replied: 'the amount of useful contribution the person can still make to the economy!!! He was further asked, " How can we dtermine that"? The expert remained silent.

7.2 Methodology for Modelling:

Two types information are used in modelling.

(a) Knowledge and insight about the system being modelled.

(b) the system inputs and outputs.

In utilising the first type of information, using deductive reasoning can generally lead to unique solutions of underlying problems.

Deductive reasoning is going from known principles to deduce the unknown. In modelling deductively, we derive mathematical model analytically making use of physical laws like Newton's laws of motion, Maxwell's equation, Kirchhoffs' laws of conservation of energy, etc. The application of these laws as appropriate, permits derivation of ordinary or partial differential equations or algebraic equations. These equations are either solved in close analytic forms or numerically. Where neither form of solution is possible, one can the result to simulation. We shall return to this topic later.

(b) The second type of information is availability of empirical data consisting of observations about the system inputs and outputs.

Inductive reasoning is used in developing model form empirical data. Given a set of data one can develop several models satisfying the observed inputs - output relationship unlike the method of deductive

reasoning which leads to unique models. Modelling using inductive reasoning requires that the model be validated before it is used for simulation experiments. In one of our works Abass (1988), we have developed criteria for model discrimination. Box and Jenkins (1970) have suggested the use of principle of parsimony: i.e. if we have more than one model, that are considered satisfactory for a given set of empirical data, we choose one that has the least number of parameter. We have shown that this can be operationalized by using Akauke Information Criteria (AIC) Abass (1968).

Apart from the problem of non-uniqueness of models, there are questions associated with the quality and quantity of data. In our experience, modelling has been further complicated here in Nigeria, because of non-availability of data, where data are available, they are unreliable, a situation further worsened by gaps in the data. In recent times,to deal with the problem of uncertainty associated with modelling empirical data, further assumptions are made about the nature of statistical distributions that could have generated the observations. This is easy to do if one has insight into the real-life system.

It is therefore very important that models derived from empirical data are evaluated and validated before they are used for experimentation. We return to this aspect of modelling technique later. If we represent modelling problem as a black - box problem, we can use Fig (2) to show the degrees of difficulties involved as we move from one end of spectrum to the other. At the end of the spectrum, the white box, are mathematical models arising in electric current theory. Here, we

usually know the structure of the circuit and most, if not all of the element values. Using Kirchoff's laws and similar network theory, one can construct a mathematical model virtually without resulting to any experimental data. Proceeding along the spectrum, one encounters problems in mechanics, such as aeroplane, vehicle control (landing pathfinder robot on the planet mars, or landing man on the moon).

Here the model is largely derived by applying laws of physics and knowledge of characteristics and dimensions of the system. From there we encounter problems in chemical process control. Here chemical reaction equations and reaction rates are used to derive models, however a number of parameters are not capable of being specified directly. If we proceed further in the direction of the black box we encounter business models, environmental models, life sciences and Social sciences zones if we move, things become less clear and there become progressively greyer until we get to the black-box. Whereas in the environmental systems, we know the chemical and physical processes involved (eg movement of water underground, reservoirs, diffusion of pollutants etc), the field within which these processes occur is not accessible.

Close analytical solutions are obtainable for models at the white box end of the spectrum. Models of aerospace, vehicle control chemical systems may require numerical solutions using the computer. Others near the black box will resume simulations.

(1) Abass, Dawoud & Seriki (1978), Abass Dawoud (1979), Abass, Ogbuja (1988), Abass, Dawoud (1980) have developed models for speech signal predictions and transmission. Prof Seriki collected the sample of speech. The efforts here were directed towards developing models for speech input into the computer.

(2) Abass (1979) has developed a model of the Nigerian Educational System. This model is an integration of three other models.

(a) A forecasting model: This is used to forecast enrollment into the primary schools in Nigeria.

(b) A transfer function model is developed for transition from primary to secondary sub-systems and from secondary to the University sub-systems.

(c) Within each sub-system, primary, secondary, and University, Markov models are developed to study the movement of students from one year to another.

7.3 Model Validation

A mathematical model as defined earlier is an abstraction of real-life systems. Before it can be certified for experimentation, it is necessary to validate the model and determine the extent to which it represents the real-life system. The more its mimics the real-life system the better, and the more credible will be our simulation results, hence the ckearer our visions of reality.

In the physical sciences, experiments can be easily performed to validate models. In the 'soft' sciences where inductive method is used to build models or statistical assumptions are made about the empirical data used to construct statistical tests are often used. One such test

is the goodness of fit test. Others are probability plots, and quantile plots. The results of these tests must be interpreted with care.

Objectives for building a model are often of great use in validity a model. In ecological models, for example, a model that supports sustainable development in the sense that it protects the environment, will be prefered over one that are statistically acceptable but less supportive of sustainable developments.

8.0 SIMULATION

Computer Visions of reality are generated through the process of experimentation on the computer with a mathematical model. In order words, the computer serves as our laboratory. This is done through simulation. What is Simulation?

The word "simulation" came from the Latin word "simulatus". Webster Dictionary defines simulation as feigning; the art of feigning, pretence, false representation. The dictionary definition of the word suggests that simulation involves imitation, mimicking, giving appearance of effect of some elements. These definitions are rather too general for our purpose.

We shall therefore define simulation as experimentation with a mathematical model of a real-life system on the computer under dynamic conditions over long periods of time. As shown in Fig (i) , the range of real-life systems is wide, covering from hard sciences to the soft sciences.

From this definition the following issues emerge:

(a) There must be a mathematical model to experiment with. In the last section, we dealt with the methodologies for mathematical modeling. We have developed mathematical model of several real-life Nigerian systems as mentioned earlier.

(b) Since experiment will be performed the experiment must be designed statistically. If this is not done, the process of data collection will be very difficult. Different designs of experiment have different methods of data analysis. Hence data analysis becomes easy. One of the assumptions when modelling is that of selectivity. By designing our experiments we control which variable will come into our experiment and which will not

Abass (1987) thoroughly examined the statistical aspects of simulation experiments so that appropriate statistical tools are emp ved when designing simulation experiments and analysing what collected from simulation experiments. We have shown in the same paper that except appropriate statistical designs are used, conclusions reached from such experiments may be very misleading.

(c) Since the experiment will take place under dynamic conditions for long periods of times, the experiment must be performed on the computer. There must therefore be a simulator for carrying out the simulation. A simulator is a program. Simulators are developed using special simulation languages like GPSS, SIMPSCRIPT, SIMPLAN, which are 4th Gls. Whereas they are very easy to use and produce results fast, we have always developed my own simulation programs First, the 4Gls languages are problem oriented: GPSS is developed for simulating queueing systems, hence it cannot be used for

simulating econometric system, for which SIMPLAN is developing. This luxury of having different simulators cannot be enjoyed in our environment where almost everything in our universities is in shortfall. Secondly, by developing one's programs, one becomes more and more skillful in programming and consequently better prepared for the softwware industry, an industry which could be exploited to develop software which can be marketed outdide Nigeria, and therfore contribute to our capacity to export.

(d) Data must be collected and analysed. The type of design we choose will determine the variables on which the data must be collected, and the method of analysis, and the duration of the experiments. Data are analysed to reveal different aspects of real-life systems.

9.0 VISIONS OF REALITY:

Simulation on the computer permits the model to be manipulated by;

(a) modifying assumptions
(b) adding new variables
(c) deleting some variables
(d) expanding the boundary of the system under study.

As these changes are made new experiments are performed through simulation, this leads to new sets of data which when analysed give us different visions of reality. These visions serve as instruments for explaining the past and present for predicting/forecasting planning and controlling the future. Let us ask one question, what is Vision 2010 all about? I believe it is to explain our past and pressent and predict, plan and control our future. What we have presented this evening is our alternative to *Vision 2010*.

10.0 IS "COMPUTER SCIENCE" A "SCIENCE"?

Mr. V.C., at the beginning of this lecture, we sought to answer the question whether "Computer Science" is really a "Science." To provide answer to this question, we proposed that a discipline will be considered as a science if its process of inquiry exhibit the following characteristics:

(a) Questions can be answered and problems can be solved.
(b) More effective ways of answering questions and solving problems can be evolved.
© Controls can be exercised.
(d) Experiments can be performed.

In section (2.1), that deals with foundations of Computer Science, we have shown that Turing Machine can be constructed to solve any compatible function provided the machine is provided the machine is provided enough time and auxillary storage. In section (3), we showed how a machine can be instructed to solve problem. Consequently characteristic (a) has been exhibited.

In section (3), (4), & (5), wer discussed the evolutions of 5th generations of programming languages, 5th generations of operating systems, and 5th generation computers. One principal aim of this series of evolutions is to ensure more effective procedures for answering questions and solving problems. Hence characteristic (b) has been displayed beyond doubt.

In section (7) above, we have shown how experimemts can be performed in the disciplineof Computer Science through simulation. By varying assumptions made while constructing the mathematical model used for simulation, we exercise control in experimentation. Thus characteristics © and (d) have been exhibited beyond doubt. It can therefore safely be inferred that the discipline of "Computer Science " is indeed a "Science".

11.0 APPROACHES TO PROBLEM SOLVING:

In the older sciences of Physics, Chemistry, etc., the approach to problem solving is mechanistic and analytic. The philosophy of problem solving is reductionism. This encourages decomposition of a problem to a number of parts, electrons, atoms, etc. Solutions are

then sought to these components and integrated. The synthesised solution is the referred to as the solutuion to the problem on hand. Aristotle, thousands of years ago, has told us that the sum of the parts is not equal to the whole.

In Computer Science and new Sciences, approach to problem solving is generally wholistic. The philosophy of problem solving is that of expansionism. In this approach, the system (problem) under study is regarded as either a sub-system of super-system. In this approach, it is not permissible to pigeon-hole science as Physics, Chemistry or Biology etc. They should be regarded as sub-systems of Science.

12.0 IDEALS OF MANKIND :

Since the dawn of recorded thought, Philosophers have sought to identify the ideals of mankind; that is the ultimate objectives to which mankind seeks closer and closer approximations without expectations of complete attainment.

Ancient Philosophers have identified three such ideals:

(a) Truth
(b) Goodness
(c) Beauty

Modern Philosophers have identified a fourth;

(d) Plenty i.e economic abundance

Mr. V.C., I wish to submit that applications of Computer have been Contributing to the attainment of these ideals.

13.0 CONCLUSION

Mr. V.C., the two tasks we set for ourselves this evening, I believe have been performed. It is now left for me to thank you and others for sharing this wonderful and glorious evening with me.

REFERENCES

1. Abass O., Dawood, Seriki (1978) "Prediction Design Using Aotoregressive Models." 13th Conference on Statistics, Computer Science and Operations research; pp 393 - 405.
2. Abass O, Dawood (1979) "A New Pitch Detection Algorithm." E. C. J. Vol 6; pp 51 - 68.
3. Abass O. (1981)"A Mathematical Model for Educational Planning." Introduction to Educational Planning, edited by Segun Adesina; pp 51- 64; published by University of Ife Press.
4. Abass O. (1980) "A Simulation Model for Electricity Consumption in Nigeria."Ricerche di Automatic,Volume II, No 2, Dec. 1980.
5. Abass O., Ogbujah J.C.T. (1983) "A System for Modelling Epidermiological Data. "MD-ISNFOR (1983), Parts 1 & II, edited by J. V. Benomol, M.J. Bell and O. Wigerts, published by North Holland; pp 1230 - 1233.
6. Abass O., Ogbujah J.C.T. (1983) " A Technique for Speech Signa Prediction and Transmission." ASME, Vol 5, 1983, pp 315 - 325.
7. Abass O. (1987) "A Semi-Markov Model for Manpower Planning." E.C.J. Vol. II, No.3, 1987; pp 18 - 21
8. Abass O. (1987) "Statistical Aspects of Computer Simulation." JORSON Vol 1, 1987,pp55-67.
9. Abass O. (1988) "On Criteria for ARIMA Model Selection. "J.C.S. Vol. 5, No.1,1988; pp 13 - 26.

10. Ackoff R. L., Gupta S. K., "Scientific Method." Minas J. S. (1968) Published by John Wiley NY.
11. Bohm D. (1957) "Casuality and Chance in Modern Physics. "Published by Rontledge and Kegan Paul Ltd. London.
12. Born M. (1957) "National Philosophy of Cause and Chance. "Clarendon Press, London.
13. Chomsky N. (1959) "On certain formal Properties of Grammars. "Information and Control; pp 137 - 167.
14. Dewey J. (1938) "Logic: Theory of Inquiry." Published by Henry Holt & Co, N.Y.
15. Dingle H. (1953) "Scientific Adventure."New York Philosophical Library.
16. Knuth D. E. (1968) "The Art Of Programmimng." Vol 2; Fundamentals of Algorithm, Addison-Wesley, Reading, Massachusetts.
17. Krishnamurthy E. V. (1990) "Introductory Theory of Computer Science." E.W. Press, Madras.

University of Lagos Press Tel: 825048