

57/26
L
36
L25
96
2001

**COMPUTING CULTURE:
THE STATE OF THE NATION**

U.D. ARCH

By

J.O.A. AYENI



UNIVERSITY OF LAGOS PRESS - 2001
INAUGURAL LECTURE SERIES

COMPUTING CULTURE: THE STATE OF THE NATION

UNIVERSITY OF LAGOS
LIBRARY

Joshua Ojo Akindele AYENI

B.Sc., PGD (Lagos), M.Math., Ph.D. (Waterloo), FCOAN, FCPN

Professor of Computer Science

University of Lagos

U. L. ARCHIVES

University of Lagos

Printed by Univ Lagos Press 2001

J.O.A. Ayeni, 2001

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the author.

ISSN 1119 - 4456

Printed by University of Lagos Press

INTRODUCTION

What is computing? Or let us begin with computer science. What is computer science? That is a question many experts have endeavoured to answer in the past and in recent times. Some have preferred to describe what computer science is about rather than give it a definition [Hamming (1968), Longe (1983), Denning *et al.* (1989)]. There is no doubt that a proper understanding of what computer science is would enhance general understanding of what computing discipline is all about and its culture.

The frequent discoveries involving the use of computers in the society are enough to confuse the uninitiated mind as to what computer science is really. Is computer science the same thing as computer technology, computer engineering, information science, information technology, Internet, computer studies? Not exactly.

Computer science is the body of knowledge and practices used by computing professionals in their work. The discipline is also variously called computer science and engineering, computing or informatics [Ralston (2000)]

The computing profession is the people and institutions that have been created to take care of people's needs in information processing and its coordination through world wide communication systems. The profession contains various specialties such as computer science, computer engineering, software engineering, information systems, domain-specific applications and computer systems.

Computing discipline is characterized by three recurrent themes: theory, abstraction and design [Denning *et al.* (1989)].

Both computer science and computer engineering agree on fundamentals and are concerned with all elements of information processing systems, including the application environment, paradigms for representing information, paradigms for processing information and tools for designing and measuring the effectiveness of computer systems. However, design is the pre-eminent domain of computer engineering while theory is central to computer science. Through abstraction, both disciplines master complexity.

UNIVERSITY OF LAGOS
LIBRARY!

The term information science was coined to designate an interdisciplinary field initially concerned with the exponential growth of recorded scientific information. Evolving from its roots in librarianship and documentation, information science has been predominantly occupied with information "content". The portfolios of concerns of information scientists today include issues of intellectual property, individual information privacy, equity in information access, etc. There is agreement that, as yet, a scientific basis for a science of information remains to be developed.

Information Technology (IT) is a term frequently applied to a broad area of activities and technologies associated with the use of computer and communications, but generally implying the application of computers to storage, retrieval, processing and dissemination of data. Thus, IT profession has an enormous scope including science, engineering, business and broadcasting.

In some usage of the term, IT can substitute for computer, e.g., IT software means computer software while IT products include computers, data communication and data broadcasting equipment. Also IT service is any service which results from the use of IT software over a system of IT products for realizing value addition [GI Task Force (1998)]

The "IT family" as coined in Ralston (2000), consists of professional groups in different sub-areas of computing discipline, some of which see themselves as peers of computer science within the IT profession and are hesitant to identify with computing. We now have IT specialist lawyers, firms with IT departments, governments having ministries of IT, universities establishing IT faculties, IT journalists are legion and self-styled IT professionals/consultants abound.

Information Technology may not survive as a well-defined discipline like computer science or civil engineering largely because of its broad scope, however, it should survive as an approach or methodology like Science or Art. It is in this light that a faculty of Information Technology is very welcome in the universities where there is one and it is expected that computer centers in the universities may become Information Technology Centres which would incorporate IT services in their activities.

MacMillian Oxford Dictionary defines culture as the "sum total of the beliefs, accomplishments and behaviour patterns of a group of people, acquired by members of the group through social learning and transmitted from one generation to another; intellectual and artistic content of this".

In the first part of this lecture, we will be discussing the beliefs, accomplishments and behaviour patterns in computing since the birth of the discipline in the early 1940s. My own humble contribution will be presented in the second part. The final part will consist of computing culture in Nigeria, conclusions and recommendations.

ACCOMPLISHMENTS IN COMPUTING

These are in the areas of hardware, software, computer architecture and Internet.

HARDWARE SUB-SYSTEM

Hardware is the physical part of the computer and it consists of the memories, storage devices, peripherals, routing and control circuits.

Memory

Memory or storage sizes are measured in bytes, kilobytes (1024 bytes), megabytes (1024 kilobytes), gigabytes (1024 megabytes), terabytes (1024 gigabytes) and petabytes (1024 terabytes). These are respectively abbreviated as B, KB, MB, GB, TB and PB.

Technologies used in realizing the main memory of the computer had advanced from the non-volatile magnetic (ferrite) core (which is now obsolete) to the volatile Metal Oxide Semiconductor Field Effect Transistor Dynamic Random Access Memory (MOSFET DRAM) used for main memories in modern times. Other technologies used are MOSFET SRAM and bipolar SRAM but MOSFET DRAM is currently the dominant technology. By 1999, 64M bit DRAM chips were available as against the 64-bit bipolar SRAM chips used in 1960s by early computers such as IBM system 360. It is projected that DRAM chips will contain up to 256G bits by 2012.

The Holy Bible, King James Revised Version, contains less than $5\frac{1}{2}$ million characters. It is remarkable that the memory size of some present day microcomputers/servers is as big as 256MB. This means that such

microcomputers can store in memory 40 different books each of the size of the Holy Bible simultaneously. This is a great achievement bearing in mind that the mainframes of the 1950s with 512K memory size cannot accommodate a single copy of the Holy Bible in the memory all at a time!

Storage Devices

Storage devices cause information to be stored for later use in the computer. Over the years storage devices have become more sophisticated and varied. Storage devices now have fast access time, large capacity and are low cost. A disk-based mass storage unit now has a capacity for up to 2TB (2^{31} bytes) costing just 9 cents for 1MB. The cost is even lower for DVD-RAM. These mass storage devices have up to 10ms access time (this is 100 memory accesses in one second) and 1MB/sec transfer rate.

Peripherals

The peripherals are devices that change information from one physical form to another and link the computer with its environment. Peripherals include keyboard, video screens, printers and various input devices such as mouse, button, joystick and lightpen.

A major accomplishment in development of peripherals has to do with technology requirements for multimedia computing applications. Multimedia computing implies the integration on computer system, of audio, video and images with the traditional types of data such as text and numerics and is based on the evolving ability of computers to store, transmit and convey diverse types of information. Some multimedia computing applications are: online news, advertising, electronic mail, banking, distance learning, digital libraries and medical information systems.

The technological requirements of a Distributed Multimedia Information System (DMIS) concern workstation technology and include: at least a CD-ROM at the workstation for assemblage of data for retrieval, communication protocols, bandwidth, internetworking, data storage, application interfaces and authoring tools, information modelling and retrieval. The high-resolution monitors and audio output are being used as presentation devices while data capture devices include: scanner for still images, microphone with digitizer for voice data, keyboard for text and camera with digitizer for video.

Work is currently going on to recognize voice and handwritten data entry and will undoubtedly lead to new devices. Personal Digital Assistants (PDA) allow input of a constrained set of hand printed characters. Data tablets are available for personal computers. These data tablets are essentially "electronic yellow pads" on which the user may make sketches and handwritten notes. Small video cameras are now available which allow two similarly equipped network users to converse through phone-picture.

Routing Circuits

The routing circuits (including buses and networks) direct flow of information between various functional parts of the hardware subsystem, that is, the main memory, the central processing unit and the storage devices. In the mainframe and minicomputers, we have buses that are capable of transferring and receiving data at the rate of up to 4.5MB/sec. The 32-bit Extended Industry Standard Architecture (EISA) bus developed by the gang of nine – Wyse, ASI, Tandy, Compaq, Hewlett-Packard, Zenith, Olivetti, NEC and Epson has data rate of up to 33MB/sec.

SOFTWARE SUBSYSTEM

Programs that are needed to make computers perform their intended tasks are called software. They include operating systems, programming languages and graphical user interfaces (GUIs).

A complete software system will contain, in addition to the operating system, a set of compilers for various languages, one or more system loaders, one or more database systems, sets of utility routines, special and general-purpose debugging systems and generalized subsystems for applications such as sorting and merging, mathematical programming, engineering design, report generation, simulation graphics and so on.

Early operating systems were control programs that were a few thousand bytes long. They scheduled jobs, drove peripheral devices and kept track of system resources usage for billing purposes. Modern operating systems are much larger, ranging from hundreds of thousands of bytes for personal computers, e.g. MS-DOS, XENIX etc. to tens of millions of bytes for main-frames, e.g. Multics, MVS, UNIX etc. and hundreds of millions of bytes for some servers, e.g. Microsoft Windows NT. In addition to managing processors, memory and dozens of input/output devices, modern operating systems also provide interprocess communications,

file and directory systems, data transfer over local networks, command languages and graphical user interfaces for invoking and controlling programs. An operating system does not only perform allocation of computing resources among processes contending for them, it also serves as an extended machine that provides a powerful programming environment.

Up till about 1969, it was generally assumed that the purchase or rental of computer hardware system entitled the customer to all general-purpose software produced by the manufacturer for that computer at no cost. In June, 1969, IBM announced that it was introducing a new policy of not supplying computer software along with the computers sold or rented with the exception of essential operating system software. Almost all other hardware manufacturers followed the lead of IBM and unbundled their software products. We now have several organizations which specialize in producing software for use with computers even though they do not manufacture hardware. Microsoft, Oracle, Magic software Systems, Dbase Inc. are examples, just to mention a few. Bill Gates, founder of Microsoft was rated the world's richest man (*The Guardian*, September 23, 2000) in year 2000 with an estimated asset of \$63 billion. What is Nigeria's gross domestic product in 2000? The 1999 estimate was \$110.5 billion! [*The World* (2000)].

COMPUTER ARCHITECTURE AND INTERNET

Computer architecture has to do with the hardware structure of computer systems, the attributes of their various parts and how these parts are interconnected. One important achievement in this area is the issue of compatibility between computers. A family of computers that share a common system architecture allows a software written for any model in the series to operate correctly. This is realized by allowing various members to share the same instruction set, arithmetic registers, input/output methods, memory addressing mechanisms, etc. In certain instances, new members of the series have extended the architecture in an upward compatible way, that is, software written for the older series can run on the newer ones, but not always vice-versa.

The Internet is the interconnection of computer networks world-wide and its realisation can be considered the greatest achievement in the computing profession in recent years. It has made the world to become a global village. It allows information to be passed from one network and

communication media to another in digital packets. The physical layer of the Internet connects users on telephone, satellite and cable TV network; local area networks (LANs) and wide area networks (WANs).

The Internet Protocol (IP) is used to route the digital packets in an efficient and generally reliable manner. The Transmission Control Protocol (TCP) manages the transmission and receipt of the entire messages between two points on the network using the IP packets. The application protocols such as Email, World Wide Web (or simply Web), Search engines e.g. Yahoo!, e-commerce, intranets and extranets are built on the TCP.

BEHAVIOUR PATTERNS

Pattern is a model or way in which something happens, moves, develops or is arranged. Behaviour is a way of acting or functioning. Thus, we can infer from the above that behaviour pattern in computing is the way things are happening in the various sectors of computing discipline and/or how the various sectors have developed or are developing over time. The relevant sectors are the hardware and software sectors.

HARDWARE SECTOR

Until about 1968, most computers were mainframes that were expensive and required considerable floor space, electric power and air-conditioning and so were used as shared systems. However, due to the rapid advances in computer technologies, especially conductor circuitry, by the late 1960s, a genre of small size machines termed minicomputers was produced. The machines occupied only the space of an office desk, and were inexpensive enough to be used for the solution of a single problem, such as the control of a particular industrial process as well as a variety of shared uses.

By about 1973, further rapid advances in technology led to another class of even smaller, cheaper systems based on the microprocessor – a complete CPU contained on a single or very few semiconductor chips. This method of manufacturing eliminated most hardwiring and separated component manufacture of the CPU with substantial improvements not only in cost but also in reliability. This great advance in technology was matched by corresponding advances in semi-conductor memory that allowed compact, high-capacity main storage to be offered at a price hundreds of times cheaper per unit of stored information than was available only a decade earlier.

By 1990, several million microprocessor-based systems, called personal computers or workstations were in common use in homes, offices and businesses. Each gave to an individual user computational power and main storage that had been available in shared form only from large expensive systems just a few years earlier. Furthermore, the progress in technology that has given us the microprocessor has also enhanced the cost-effectiveness of the large minicomputer and the general-purpose mainframe systems.

SOFTWARE SECTOR

From the outset, the development of software has been directed toward the apparently contradictory, but in fact complementary goals of bringing the computer closer to the user while keeping the user at a distance. The first goal had involved the creation of programming languages and systems to facilitate the development of the application that make the computer useful. The second goal had included the operating systems that oversee these applications and manage the hardware and software resources on which they draw. The third and present goal has been in interposing layers for constructing large software efficiently.

Programming Languages

We can define a language as a particular system of words, symbols, signs etc. used for conveying information. Then such system as used by the English people is English language and the one used by the computer professionals to direct the operations of the computer is a Programming language.

Development of languages started as early as 1945 with the design of a high level programming language called PLANKALKUL by Konrad Zuse [Bauer (1972)] in Germany but was never implemented. Autocode, the first working high level programming language was developed by Alick Glennie in 1952. However, the first operational compiler was developed by Lanning and Zierler for mathematical computations and implemented on MIT whirlwind computer in 1954 while John Backus' Fortran compiler was most widely used in mid – 50s.

Early languages maintained dichotomy between scientific calculations and data processing. Early examples are Fortran (1956) for scientific calculations and Cobol (1960) for business data processing. With the advent of interactive computing, languages like Joss (1963) and Basic (1964) were developed for effective use in an interactive environment.

Between mid-1960s and late 1970s, languages were developed for multipurpose (or general) processing. These include PL/1 (1964), Algol 68 (1968), Pascal (1971) and C (1973) for system programming. More powerful languages suitable for embedded computer systems (i.e. systems which involve computer and other equipment) were developed in the 1980s and 1990s. These include Ada (1979), C++ (1980) and Java (1995).

Standardization of programming languages started in the early 1960s with the American National Standards Institute (ANSI) coming up with the first Fortran standard X3.9 in 1966. Standardization aims at decreasing the amount of conversion required to move a program from one environment to another. Subcommittee 22 (SC 22) of ISO/IEC (International Organization for standardization / International Electro-technical Commission) Joint Technical Committee 1 (JTC 1) is an international body which is responsible for international standardization of languages for information technology. It has working groups for various languages like Ada, APL, C, C++, Cobol, Java, etc. in different countries.

The following language concepts have endured: formal method for defining language syntax (BNF: Backus – Naur – Form), block structure (Algol), data typing, record structure and separation in a program of data and procedural aspects (Cobol), class concept (Simula) which has led to modern concepts of data abstraction and object-oriented programming, strong data typing mechanism (Pascal) and conceptual autonomy (Java). Java includes in its design the specifications for the virtual machine on which it runs.

Operating Systems

In the early days of computing, the programmer tended to operate the computer alone. He would mount the tape, put punch card into card readers, etc. This resulted in inefficient use of the programmer's time. Also, the computer processor was having too much idle time since it has to wait for the programmer to give instruction to continue each time it paused. About 1956, a simple operating system was developed jointly by General Motors and North American Aviation for the IBM 704. This system provided facilities for sequencing from one job to another, input/output control system, calling compiler, loading jobs, etc.

By the 1960s, large and powerful batch operating systems, e.g., OS/360 for IBM computers were developed. They provided facilities for multiprogramming.

By early 1970s, UNIX became widely accepted as a model of small but powerful interactive operating system designed to facilitate the work of the programmer.

With the arrival of interactive systems, more people, many of whom are non-computer specialists became interested in using computer. Hence, attention was turned to providing suitable user interface. Earlier, interfaces consisted of jumper wires in patch boards, punch cards, prepared off-line print-outs etc. Today, the user interfaces include keyboards, mice, graphical displays, etc.

The rapid expansion of memory and processing speed dedicated to single, stand-alone use but with opportunity to access other computers in a network led to the development of operating systems anchored on graphical user interfaces (GUIs) or WIMP (Windows-Icons-Menus-Pointers), the latest of these being Microsoft Windows 2000. The use of mouse – driven graphical user interface was popularized by the introduction of Apple Computer's Macintosh in 1984 having been inspired by earlier research at XEROX PARC.

Software Development

Programme is a list of planned activities to be carried out. In a similar vein, software or computer program is a list of instructions/activities to be executed/performed on a computer. [The act of putting together a program(me) is called programming].

In the past, programs were thought of as instruction sequences. Software production then was on ad-hoc basis employing subroutines or subprograms. Constructing large software was very expensive in spite of the relatively inefficient programs that were produced.

Following Edsger Dijkstra's letter to the Editor of the Communications of the ACM in March, 1968 entitled "Goto Statement Considered Harmful" [Dijkstra (1968)], research was intensified in constructing programs using a methodology called Structured Programming (SP). This methodology avoids using "goto" in designing programs as much as possible. In its

place, control structures were used. Bohm (1966) actually shows we can construct such programs by using control structures to link parts of a program together logically. Such programs are believed to be more readable and maintainable. They are also more likely to be correct, as parts of the program can be shown correct by systematic program verification. The acceptance of structured programming philosophy by professionals is reflected in the wide use of structured languages in teaching programming and language design in the profession.

Since the 1980s, however, object-oriented programming approach has become a prevailing form of software design. It packages data and operations into objects that have simple interfaces through which they interact. The approach is intended to manage complexity and facilitate software reuse by clearly separating interfaces from the implementation details.

BELIEFS

Let us recognize two broad classes of beliefs in computing – those beliefs which translate into professional norm or tradition (spiritual beliefs) and those that model or shape future activities in the profession in some way (technological). We expatiate further.

NORMS

Computer System Development Life Cycle (SDLC)

Conventionally, computer – based systems are implemented in many stages depending on the complexity of the system. These stages include:

- Investigation and detailed study of existing system;
- Analysis and design;
- System development (programming);
- Testing;
- Change – over;
- Maintenance and Review.

The above stated stages constitute a life-cycle of the computer-based system in that the process can be repeated all over again to obtain a new system if the existing system, at the review stage, is found to have become inefficient, inadequate or problematic. However, the new approach to developing computer-based systems using rapid application development tools like Microsoft Visual Basic, Magic, Oracle, etc. cause the above

stated stages to be applied repeatedly at every stage of the new approach which generally consists of 3 stages viz: analysis, prototyping and deployment especially at the prototyping stage even before the final product is deployed. One obvious advantage of the new approach to developing applications is the greater end-user involvement in the development of the new system.

Cost-benefit Analysis

Also, it is conventional in computing to base the justification for using computers to solve organizational problems on the versatility of the computer rather than on cost minimization. The use of computer requires substantial investment in hardware, software and manpower (humanware). Thus, the gain in using computer to solve problems is not necessarily in the amount of naira or dollars that might be saved in getting problems solved, it is rather in its cost-effectiveness, that is, in getting value for additional money being spent. However, results are usually obtained quickly and timely and as accurately as possible when computers are used.

Creativity

Computer professionals have tendency to be creative just like an artist. Sometimes this feat is achieved during hacking. Hacking is a programming style in which programs are created in an unconventional (non-systematic) manner. Hackers believe that programs should be built "straight from your mind" [Turkle (1984)]. They prize concise, efficient, elegant and even tricky code. Although more conventional programmers frown at this programming style, many hackers-developed projects such as operating system Linux and text-editor Emacs are premier examples of the programming craft. Many of the founding fathers of modern computing are former hackers – Bill Gates of Microsoft and Steve Wozniak of Apple Computer fame are prime examples. On the dark side of hackers are those who create programs called virus (and worms) to damage computer systems. Love Bug infected 45 million computers in twenty countries within 24 hours in 2000! [Ralston (2000)].

TECHNOLOGICAL BELIEFS

Here are some predictions made as a result of technological developments in the field of computing in recent times.

Software Market

Component software is designed to solve programming problems in small, modular bits, rather than in large, functional programs. Eventually, it is possible and likely that we might see components becoming an alternate software market or even a replacement software market.

100th Edition of Encyclopedia in 2768

The first edition of Encyclopaedia of Computer Science was released in 1976, the second edition in 1983, the third edition in 1993 and the fourth edition in 2000 – an average of 8 years in between editions. It is believed that the 100th edition will be released in 2768.

7th Generation Computers to be announced in 2013

In 1991, Japan announced the coming of the fifth generation computers based on Artificial Intelligence. In 1997, Japan abandoned fifth generation computers and announced the coming of the 6th generation computers based on Neural networks. It is believed that Japan would abandon the 6th generation computers in 2013 and announce the coming of the 7th generation computers.

Intel to release Pentium XCI in 2095

In 1993, Intel Corporation announced Pentium chip (Intel 80586) which was, however, flawed because for certain operands, quotients of floating-point divisions were accurate only to 3 decimals rather than the normal 16. Pentium II (Intel 80686) was announced in 1997. Pentium III was announced in 1999. It is believed that Pentium XCI (1991) will be announced in 2095.

Do you believe all these?

U. L. ARCHIVES

Other predictions

Windows 9.5K will be released in 9500

Work will begin on Y10K problem in 9999 all over the world.

It is now expedient for me to go to the second part of the lecture

CONTRIBUTIONS

The focus of my research over the years had been in the involvement of computers in operations research activities, program compilation, games, information systems and artificial intelligence. I would now delve briefly into what my contributions are in these areas.

OPERATIONS RESEARCH (OR) TECHNIQUES

OR methodology came into limelight after World War II (1945) when the British recorded some successes in the use of the method. The Americans decided to capitalize the approach by setting up a department in the Ministry of Defense to carry out research in this area. G.B. Dantzig first came up with the Simplex Method to solve linear programming problem in 1947. Linear programming is the first mathematical model that was applied to solve problems of the firm under various constraints such as scarcity of funds, materials and manpower; uncertainty, etc. Similar models that have been researched into over the years and applied include Forecasting, Inventory Control, Dynamic Programming, Queuing Theory, Game Theory, Simulation and so on. All these models are usually referred to as operations research models.

Search Methods

A large proportion of our day-to-day activities at home, at work, on the computer, etc. involves searching and sorting. In fact, solution to any problem requires one form of search or another. The search can be carried out with optimization algorithm or a heuristic algorithm. In the former case, the best possible results under the given circumstances are obtained while in the latter case good solutions are found with significant computational savings.

In certain situations, one has reliable information about the solution to a problem. It is expedient for someone to exploit such information to an advantage in any search strategy. In a research project supported in part by the National Research Council of Canada under Grant A4124, Ayeni and Magazine (1979) developed a heuristic algorithm that makes use of a prior information in finding the maximum at location x^* of a single-variable unimodal function $u(x)$. The functional form of $u(x)$ is not known, but an experiment can be performed for any $x' \in [a, b]$ and $u(x)$ determined. Alternatively, $u(x)$ may be given but is too cumbersome to take advantage of its structure (other than unimodality).

Let us represent the prior information as a probability distribution over the interval to be searched. Define $S_N = (x_1, x_2, \dots, x_N)$ as any sequential search strategy with N experiments and $?_N(G)$ as the class of such strategies that make use of a given prior information, G .

Let $L_N(S_N)$ be the interval of uncertainty which remains when strategy S_N is used. We wish therefore to find S_N^* such that

$$E(S_N^*) = E(L_N(S_N^*)) = \min_{S_N \in ?_N(G)} E(L_N(S_N))$$

When no information is given, the objective is typically to find that strategy which minimizes the largest of any possible interval. Kiefer (1953) found this strategy and called it the Fibonacci search. The strategy does not make use of any given prior knowledge. This is a drawback of a sort.

Fine (1966) showed that if $G_{x^*}(x)$ is uniformly distributed, the Fibonacci search minimizes the accepted interval of uncertainty. Shapiro (1974a, 1974b) found optimal strategies by solving a large non-linear programming problem under a restricted class of strategies. For instance, when $N = 20$, his problem requires over 10,000 variables!

Ayeni (1977) found a strategy that performed the Fibonacci search on the cumulative probability distribution G_{x^*} and called it Fibonatile search.

The search has the same expected interval of uncertainty as the Fibonacci search. The drawback of this strategy is that it considers the prior information exclusively, ignoring the lengths of the possible intervals.

Ayeni and Magazine (1979) showed that the optimal solution lies in between the Fibonacci and Fibonatile points. Ayeni (1977) has shown that this is true if G^* is convex, concave or $dG_{x^*}(x)/dx$ is symmetric about the midpoint of the interval. The observation is stated as a conjecture that is true in general.

Based on the above observation, the problem can be stated as:

Find q_i : where

$$q_i = [(1-\beta_i) i/F_N + b_i \beta_i^{-1} (i/F_N)] (b-a) \\ i = 1, 2, \dots, F_{N-1} ; 0 \leq \beta_i \leq 1$$

The heuristic search strategy simplifies the problem by assuming $\beta_i, i = 1, 2, \dots, F_{N-1}$ are stationary i.e., we wish to find

$$q_i^H = [(1 - \beta^H) i/F_N + \beta^H G^{-1}(i/F_N)] (b-a) \\ i = 1, 2, \dots, F_N - 1, 0 \leq \beta^H \leq 1$$

The proposed solution S_N^H is simply to place x_i^H at q_i^H whenever the Fibonacci search places its experiments at point

$$(b-a) i/F_N$$

The objective function becomes

$$E(S_N^H) = \sum_{i=0}^{F_N-1} (q_{i+1}^H - q_i^H) G(q_{i+1}^H) - G(q_i^H)$$

which is clearly less than or equal to $(b-a/F_N)$ since $b^H = 0$ is feasible and corresponds to the Fibonacci search.

The different measures were used to measure the performance of the heuristic search viz:

$$D(S_N^H, S_N^*) = \frac{E(S_N^H) - E(S_N^*)}{E(S_N^*)} \times 100\%$$

is used to measure how good the solution is compared to the optimal solution and

$$D(\text{Fibo}, S_N^H) = \frac{(b-a)/F_N - E(S_N^H)}{(b-a)/F_N} \times 100\%$$

is used to measure how well the prior information is used, in other words, how well the algorithm performs when compared with Fibonacci search. Some results are hereby presented.

Ayeni and Magazine (1982) show that when some prior information is given, if the cost of experimentation is considered, a myopic rule (one-step-look-ahead) stopping strategy is not optimal for sequential strategies that utilize the prior information.

In search of a best possible schedule to work with, Aimufua and Ayeni (1996) designed a heuristic algorithm for improving a university lecture timetable. Penalties are assigned to periods depending on courses scheduled for that period. A heuristic approach to minimize the sum of penalties between any two periods in the entire timetable is adopted. The algorithm terminates when no further improvement can be achieved with the sum of the penalties following two or more consecutive iterations.

The last iteration is taken as the desired improved timetable. Analysis of the algorithm shows that the computational order is $O(n^2)$ where n is the number of courses on the time-table.

UNIVERSITY OF LAGOS
LIBRARY

Mathematical Programming.

Mathematical Programming forms a large class of resource allocation strategies that are widely used in Operations Research. Mathematical Programming models include: linear programming, integer programming, goal programming, geometric programming, dynamic programming, quadratic programming and non-linear programming. Each model has its own basic assumptions and hence is applicable to problems where those assumptions hold reasonably. Linear programming model is most basic and its assumptions include proportionality, e.g., if one unit provides N5 contribution to profit, 7 units should provide N35 contribution; determinism, i.e., all parameters of the model are known constants; etc.

Linear Programming problem can be solved efficiently using any variant of simplex method discovered by Dantzig (1963). Ayeni (1983) has shown that application of non-linear programming model could be preferable to linear programming model (even though it could be more computationally difficult to solve) in some production systems. For comparison, Ayeni used a separable programming strategy that employs adaptive procedure of the Golden Section search as we have in Wagner (1975) to solve an hypothetical production problem.

In Ayeni (1980), a code for a computer algorithm for solving a transportation problem in Algol language was presented. The algorithm is a special form of Linear Programming model. It uses the stepping-stone method to determine the variable to leave the basis and one to enter the basis. Ayeni (1988a) also coded the algorithm in Fortran (WATFIV) on IBM 370/145 computer. The program was capable of solving a transportation problem that involves distribution of products from 200 origins (sources) to 200 destinations (markets).

U. L. ARCHIVES

Simulation

Determining the best time-slice (or optimal phase length) for a road junction traffic control system depends on among other things: optimal number of control phases to serve the junction and arrival and service patterns of vehicles that enter and leave the junction. In a work to investigate these factors, Ayeni (1988b) modeled the number of phases required as a colour

problem as suggested by Aho (1983). A closed-form mathematical solution for the problem does not exist for now.

The arrival pattern distribution is poisson and the service distribution is exponential. Parameters of these distributions are to be obtained from the field. A computer simulation model was implemented in BASIC programming language to determine the optimal phase length (time-slice) of a traffic light control system for a double T-junction.

The T-junction is typical of 3-access road junctions such as the one at the AP petrol station and the one at the Bookshop at the University of Lagos here. Three phases are adequate to control the junction. Data was collected for estimation of the value of the appropriate parameter and was processed. The simulation results indicate that the control system gives better performance when the knowledge (prior information) of the arrival distribution of vehicles into junction is taken into consideration while allocating time to each phase than when it is ignored. It is also found that the average queue length in the system is directly proportional to the total phase length. Details of this simulation implementation are contained in Ayeni (1990).

Game Theory

Von Neumann and Morgenstern (1947) provided the theory of zero-sum two-person games and characterized their solutions. Zero-sum two-person games are games played by two persons and what one person gains the other loses. A solution to the game is a pair of strategies in equilibrium, that is, a player stands to be at a disadvantage if he fails to adopt a strategy in equilibrium.

Ayeni and Longe (1985) characterize Ayo, a game that is widely played in West Africa. A carved wooden box (board) called OPON is usually used to play the game. The typical size of OPON is 50 x 20 x 8 cc. It contains two sets of 6 fist-deep holes dug on the board. Two additional holes (initially empty) may be dug at each end of the board for storing seeds won by each player as game proceeds. At the start of the game, 48 seeds of the plant *Caesalpinia crista* [igi ayo; see Odeleye (1977)] are distributed in fours into the 12 holes. Each player sits comfortably close to a set of 6 holes (called active houses in this lecture) facing the other player. This set of holes or houses is said to belong to him.

There are many versions of the game. We limit our discussion to **Ayoyayo** in this lecture. A player starts the game by selecting one of the active houses (holes) in the set that belongs to him. The seeds in the selected house are distributed anti-clockwise one seed into each next active house. The distribution may continue up to and through the opponent's (other player's) active houses and even back to the player's own active houses and so on. However, no seed must remain in, or be distributed into the original selected house at the completion of distribution.

For clarity of illustration, the houses had been numbered anticlockwise from 1 to 12 and the non-active houses, I and II.

If player 1 (P) selects to deal from house numbered 5, the four seeds in house 5 will be distributed into houses 6, 7, 8, and 9 in that order. If player 2 (Q) plays next as it should be, selects house 7, he will distribute the 5 seeds that are now in house 7 anticlockwise into houses 8, 9, 10, 11 and 12. At this juncture, the OPON is transformed into a new state.

A player collects (wins) some seeds from OPON if, at the completion of distributing the seeds from a house, he ends up at a house or at a set of adjacent houses in the opponent's row that contains 2 or 3 seeds each provided that the removal (or collection) of such seeds does not lead to inadvertent stoppage of the game, that is, it does not lead to a situation where the opponent does not have opportunity to continue with the game. Either, all seeds won are removed completely or none of the seeds won are removed from OPON at any stage of the game and those removed may be deposited into the inactive house assigned to the player or if no inactive house is assigned, such seeds are kept in one's fist for the duration of the game.

The game ends when any of the following events occur

- i) A player surrenders (accepts defeat)
- ii) No more than 3 seeds remain on OPON
- iii) A player causes inadvertent stoppage of the game
- iv) Too few seeds remain on OPON, thus preventing meaningful continuation of the game.

In the analysis of the game, Ayeni *et al.* (1985) show that using a one-step look-ahead strategy (considered a myopic decision) to determine which house to deal from, the resulting pay-off matrix for the game exists

and has a solution which is either a saddle point obtainable using minimax criterion or a mixed strategy obtainable using simplex method to solve a resulting linear programming problem. Even in a situation where *odu* or *ikare* concept is considered, it is shown that the solution methodology for the resulting payoff matrix remains valid.

Ayeni *et. al.* (1990) describe a Pascal implementation of the Ayo software as described above on an Apple II microcomputer. Also the work of Longe and Ayeni (1991) discuss some practical difficulties encountered in designing software for playing Ayo game. Some of these difficulties include intermixing text and graphical modes, strategic selection and efficient algorithm for configuring OPON.

PROGRAM COMPILATION

Human beings communicate instructions (that are to be obeyed) to computer (and vice versa) and computers communicate instructions to each other through one programming language or another. Spoken languages cannot be used directly because computers do not understand spoken languages directly. The programming language instructions (usually called program) are coded in binary digits (bits). Communicating instructions in coded bits are not natural to human beings neither is it easy or efficient to do so. To overcome this difficulty, programming languages that resemble spoken languages or natural languages are developed and are used to communicate instructions to and fro computers. A problem thus arises in that the instructions are generally not in coded bits.

To resolve the issue, compilers and interpreters that translate these programming language (usually called high level or very high level programming language) instructions to coded bits that can be understood directly by computers are designed. The process of translating or converting these language instructions to coded bits is called compilation. The first task of a computer in trying to obey (execute) instructions is to know what the instructions are. If the instructions are passed to the computer in a written form, the task is called scanning process and it is performed by the scanner or lexical analyzer.

Ayeni (1982) researches into problems that are usually encountered by the lexical analyzer and proffer solutions. It is shown that the lexical analyzer could ignore blanks (spaces or gaps) in instructions in many

situations and still be able to produce correct transformation or tokens for the instructions. Note that if the computer does not pick the tokens correctly, it could be obeying different instruction from what the user has intended. Use of regular expressions combined with table look-up is prescribed for obtaining correct tokens.

The compiler (or interpreter), too, is a program written in a low or high level language. Early languages such as Fortran, Cobol and ALGOL have been improved upon by designing completely new languages such as Pascal, C, C++, PL/1, Ada, Java, etc. that feature new concepts in programming languages. Thanks to automatic generators, compilers for these new languages did not take too long to realize.

Instead of designing completely new languages, existing languages can be improved upon by extending their capabilities. In this wise, the existing compilers are still used to compile programs written in the "extended languages". However, there is the need to preprocess the "extended features" to enable the existing compilers understand the "added" features. Uwadia and Ayeni (1988) extend Fortran IV language by incorporating features for writing structured programs and input/output statements that are not in Fortran IV or ANSI Fortran languages. The extended language is nicknamed LAGFOR. A preprocessor is written that pre-compiles these statements that are written in the "added" features and the output of the preprocessor is then compiled along with other source statements which do not require preprocessing by the compiler of the basic language (using the existing language compiler). Preprocessing offers much benefit with relatively little effort [Kemighan (1976)].

One other difficult problem we have in compilation process is code generation. It is the code generated that is actually executed on the computer. It has not been easy to write general code generator because the machines/computers for which codes are to be generated have different instruction sets. The instruction sets determine what code to generate for which instruction that is being processed on the computer.

One approach to solving this problem is to develop a program called code generator to accept machine descriptions as input and produce as output, code pattern tables. Since these tables are machine-specific, the generation of code for a new target machine would require describing the new machine and producing a new set of tables. Catell (1982), Graham (1982), Henry (1984) and Aho (1985) adopt this approach.

Uwadia (1990) observed that some of these machines have similar design and this similarity can be taken advantage of to further simplify the retargeting process. Uwadia and Ayeni (1994) reported a methodology for obtaining a nucleus set from the instruction sets of Zilog80, Intel 8080 and Motorola 6502.

INFORMATION SYSTEMS

Reynolds (1988) defines information system as a special class of system whose components are people, procedure and equipment that work interdependently under some means of control to process data and to provide information to users.

Mallach (1994) identifies various types of information systems. They include transaction processing systems (TPS), management information systems (MIS), and decision support systems (DSS). Others are: office information systems (OIS), executive information systems (EIS), expert systems (ES) and strategic information systems (SIS). Sprague (1980) in his pioneering work on DSS observed that DSS specially focus on features that make them easy to use by non-computer specialists in an interactive mode. They also emphasize flexibility and adaptability to accommodate changes in the environment and decision-making approach of the user.

In the research carried out by Ayeni (1994) on the feasibility of a micro-based prototype students' administration system with partial support from the Central Research Committee, it was found preferable to design the system as a distributed system that can be implemented on a network of microcomputers [Ayeni and Uwadia (1996)]

A prototype implementation of the system as an MIS is documented in Ayeni (1994).

A faculty subsystem for the students' administration system in a university environment is implemented in Dbase IV in a windows environment as a decision support system (DSS) and is reported in Ayeni and Bamgbose (1999). The software nicknamed SAS allows the following activities to be carried out on the computer among others:

set parameters e.g. grading system, degree classification etc.
view available courses in departments and programmes
enter results into system
generate reports e.g. mark sheets, broad sheet etc.

ARTIFICIAL INTELLIGENCE

The Encyclopedia of computer science [Ralston, (2000)] defines Artificial Intelligence (AI) as "a field of computer science and engineering that is concerned with the computational understanding of what is commonly called intelligent behaviour and with the creation of artifacts that exhibit such behaviour". Further exposition on AI reveals that the field can be viewed in the directions of computational psychology, computational philosophy and machine intelligence.

Machine intelligence is concerned with specific tasks and how to extend the frontier of what can be programmed on computers (i.e., knowledge) about the task. This is what reasoning is. For example, using explicitly stored knowledge to produce additional explicit knowledge is reasoning.

Njoku (2000) in his doctoral thesis under my supervision, has been concerned with the task of developing expertise in playing Jonda, an indigenous game that has rich features for artificial intelligence research. He has studied the game for the purpose of experimentation in machine intelligence. The expert system programmed in PROLOG language is a predictive model that rationally and dynamically selects slot to deal from without cause to backtracking. Njoku and Ayeni (2000) have established that in playing the game the degree of competitiveness increases slowly as the board size is increased and tends to a limit.

COMPUTING CULTURE IN NIGERIA

Acting Vice-Chancellor Sir, I have presented to this august audience computing culture and what my humble contributions had been in its various aspects. It has been shown that computing culture entails great accomplishments in the provision and usage of sophisticated data processing and data communication equipment with great potential in improving socio-economic conditions in a society. Standards are being set in products and professional practice to ensure continued relevance

and retention of the culture. We now want to address the state of our nation in this field.

Basic Infrastructures

Electricity is a scarce commodity in Nigeria in most homes and offices. At any rate, the Federal Government of Nigeria has promised abundant electricity supply for the nation by the end of 2001.

Some of the telephone exchanges are still analogue, e.g., Somolu exchange, Mainland Ebute Metta Exchange etc. The telephone services in the areas covered by these public exchanges are at best in fits and starts just like the supply of electricity in most parts of the country as at now. The International Telecommunication Union (ITU) prescribes one telephone to 100 people. Nigeria with a population of over 120m has a little over 400,000 telephone subscribers (700,000 installed capacity). This makes one telephone available to about 300 people. A new (revised) National Telecommunications Policy was approved late 2000. Perhaps this would usher in many additional fixed and mobile lines in 2001 and beyond.

Though Nigeria has been connected to the information super highway – the Internet, the facility is yet to be widely subscribed to.

Computer Hardware and Software

The number of computers in Nigeria is no doubt small. While South Africa with population of about 45m has about 25m computers, Nigeria has just over half a million computers for its over 120m people. To make more computers available, the organized private sector had taken the initiative to assemble personal computers (PCs).

Since 1997, the Computer Association of Nigeria (COAN) had been organizing a yearly Software Exhibition. Among the objectives of this activity is 'to coordinate the process of ensuring adherence to the International software development and documentation standard viz: ISO 9000 – 3' [COAN (2000)]. I believe this is a step in the right direction. It would on the long run ensure that software developed in Nigeria would have acceptance in other parts of the world.

Education

The Universal Basic Education scheme of the Federal government is yet to fully come on board. The scheme is a conscious effort to increase literacy level (which is presently 57.1% [The world (2000)], UNESCO's figure is 42%, Federal government's 52%) of the average Nigerian. Few years back, the government planned computer literacy programme that was not executed in a large scope for one reason or another.

Our tertiary institutions still have very much less number of students in science-based programmes than in non-science ones. Even though there is Federal government policy that stipulates 60:40 admission ratio in favour of science/technology-related disciplines, the Joint Admissions and Matriculation Board (JAMB) to date hardly observes this ratio. The 1998 JAMB admission statistics of candidates shows only 43% for science/technology-related programmes [Ogunleye (1999)]. The fear is that even if our country becomes highly computer literate, majority of us might end up being at best technology users. To really keep up with the world in the area of information technology or any other technology for that matter, our country should pay special attention to science.

Stake holders

The Federal government is yet to come up with an IT policy. Personally, I don't think this is a serious setback for now since the basic infrastructures are not there yet. Government can still be doing things that can enhance activities in computing. What I consider a serious problem is the lack of priority given to the Ministry of Science and Technology (MST). If Nigeria wants to be part of the information age in the twenty-first century, priority attention must be given to IT infrastructures, products and services. In budget 2001, only 0.3% of the total national budget (recurrent and capital) is allocated to the ministry.

The Federal government set up Computer Professionals (Registration Council) of Nigeria (CPN) through Decree no 49 of 1993. This body is the sole organization that is given the responsibility to regulate the practice of computing and register practising computer professionals in Nigeria. Unfortunately, the government is yet to adequately fund this body to enable it perform its lawful duties unhindered. The Computer Association of Nigeria (COAN) and the Information Technology (industry) Association of Nigeria (ITAN) have been organizing activities such as seminars, workshops and conferences to promote acceptance of computing culture in Nigeria.

Nigeria's organized private sector (OPS) is presently groaning (*The Guardian*, Tuesday, November 7, 2000) that its investment in PC assembly is facing difficulties such as heavy operating costs, unstable exchange rates, delays in clearing goods, high tariffs (10%) on IT products, etc. Unless government consciously assists this sector, nothing worthwhile may be achieved by the operators.

In the draft copy of the National Report on the Millennium (Y2K) Bug [FMST (2000a)] prepared by the Federal Ministry of Science and Technology (FMST) (I happen to be among the team of experts that prepared the report), we have it that "Computers in Federal organizations were mainly for document processing and analysis that will not have adverse effect come January 1, year 2000. Federal organizations using mainframe/mini were not many but there were serious software problems, more than hardware as some of the customized software written in Cobol, dbase and other programming languages that required a substantial amount in effecting correction". The general impression is that except for some oil companies and very few financial houses in Nigeria, most organizations use computers for document processing, payroll, billing and straight forward analysis that may be time-consuming. The computer applications in many cases are meant to serve as checks on the more convenient manual processing. The use of expert systems is not in vogue and computer networks such as client/server systems are not available in many organizations.

The MAIN REPORT in the final NATIONAL REPORT ON THE MILLENNIUM (Y2K) BUG [FMST (2000b)] summarises the state of computing in Nigeria as at present. These are the "lessons learnt from Y2K experience". They include the following:

- very low computer awareness
- very low computer literacy
- computerization at federal, state and local government levels is abysmally low or non-existent
- weak power infrastructure and poor telecommunication facilities
- slow capacity building in the education sector
- no Information Technology legislation

low level investment in the IT sector by banks and financial institutions present structure, size capacities and ability of IT companies cannot compete favourably with the rest of the world.

The above lessons show that as far as computing is concerned, Nigeria is still in the swamp.

CONCLUSION

As abstract as computing culture appears to be, one should emphasize that it is more than just providing relevant basic infrastructures and setting standards. It has got a lot to do with people, knowledgeable people, with right training and perceptions. It is in this light that the government of India [GI Taskforce (1998)] launched "OPERATION KNOWLEDGE" whose aim was to "universalize computer literacy" and also to spread the use of computers and IT in education. It was planned to set up a National Council of IT Education to initiate a 'Teach the Teachers' (3T) programme for upgrading, on a regular basis, the IT knowledge and skills of teachers. The overall target is "IT for all by 2008". To realize its vision, the government of India has earmarked 1-3% of the budget of every Ministry or Department for applying IT in the Department/sector; this investment includes not only the purchase of software and hardware products but also for training and providing IT services.

How does a nation go about cultivating computing culture? We shall consider just two cases to illustrate the approach. First, China, a developing country, uses the concept of management support systems to prepare its state-owned enterprises (the Federal Republic of Nigeria recently appointed Board members to 137 of such institutions [see *Sunday Tribune*, 12 November, 2000]) to achieve its desired goal of becoming a developed country as inexpensively as possible. Second, we will look at how some developed countries are reaping the dividends of computing culture through the application of artificial intelligence.

"The Chinese government regards the use of computer technology as one of the decisive factors in improving China's socio-economic conditions. Since the late 1980s, China has channeled sizeable resources to acquire more sophisticated methods, tools, and techniques for the development and applications of computer-based information systems (IS)" [Zhi *et al.*

(1997)]. In 1990, a guideline was issued which defined a management support system (MSS) as 'a computer-based multi-level, multi-purpose and multi-function IS'. Such a system can provide overall control of the operational and managerial activities of an organization and it can also provide information to assist managers to make decisions at various organizational levels". This type of system is a combination of transaction processing system (TPS), management information systems (MIS) and decision support systems (DSS).

In the mid-1980s to late 1980s, networks technology, especially LANs, developed rapidly in China and most enterprises gradually shifted from purchasing application software abroad to self software development, frequently jointly with universities, vendors and/or research institutes. To really get out of the swamp, companies exerted enormous efforts in the training of its managerial staff. All functional department heads learned to electronically access information that was not easily accessible manually thus realizing the huge benefit brought forth by the MSS. All hardware and software were integrated to form client/server architecture. China now has a China Wide Web on the Internet.

In the developed countries, artificial intelligence (AI) is extensively applied to deliver services to the people. Most applications of AI are of a type called expert systems. An expert system is a computer program that reasons using knowledge captured and embedded explicitly within the program to solve complex problems. I give below just a few applications of expert systems in Britain and USA [Ralston (2000)].

American Express uses Authorizer's Assistant (AA) to assist a human authorizer with the decision to allow or disallow a particular credit arrangement by analyzing a large amount of customer data from the company's database files.

British Department of Social Security uses expert system in answering the written queries of citizens concerning their pensions.

US Internal Revenue Service uses Tax payers' Assistant Expert System to give accurate tax information in response to telephoned queries from taxpayers.

The American Airlines use MOCA expert system to schedule hundreds of aircraft for varying types of required maintenance.

RECOMMENDATION

It is obvious that any expected improvement in computing culture in Nigeria would depend on what the Federal Government in particular and Nigerians in general do or fail to do. The public and private individuals and institutions have to be mobilized to contribute their own quota faithfully. I, therefore, make the following recommendations:

U. L. ARCHIVES

1. The Federal Government should give tacit and financial support to the Computer Professionals (Registration Council) of Nigeria (CPN) to enable it pursue, unhindered, the objectives for which it was set up, among which is the regulation of the practice of computing in Nigeria. Also, it should approve without further delay, career cadres for the computing practitioners in private and public services to encourage professionalism in the sector;
2. Government should plan towards setting up a world class information infrastructure with an extensive spread of fibre optic networks, satcom networks and wireless networks for seamlessly interconnecting the local information infrastructure (LII), national information infrastructure (NII) and the global information infrastructure (GII) to ensure a fast nation-wide onset of the internet, extranets and intranets. The Ministries of Science and Technology (MST) and Communications should be given this responsibility and to be able to realize this project as soon as possible, their annual budgets should be increased substantially because of the capital intensiveness of the IT infrastructure. It is suggested that up to 6% of the nation's Gross Domestic Product (GDP) should be allocated to them;
3. Tariffs on IT products – hardware and software should be zeroed until the nation can compete favourably in IT sector internationally;
4. A national scheme like China's Management Support System (MSS) should be put in place to enhance computerisation of operations in Federal, State and Local

Governments-controlled enterprises. A sub-scheme that would entail the setting up of an Artificial Intelligence (AI)/Decision Support System (DSS) unit in each Computer or Information Technology department of each enterprise may be used to trigger off implementation of comprehensive information systems in these establishments. Enterprises should be encouraged to create websites on the Internet;

5. Government should adopt more pragmatic approach to enforce 60:40 stipulated ratio in favour of science/technology – related disciplines at secondary and tertiary levels of study in our educational system. My recommendation is that special incentives be provided for people in science/technology related disciplines who are students or workers. By this way, Nigerians will be empowered to become producers as well as consumers of IT products and services of international standards;

6. There is urgent need for IT legislation. This will give legal backing for enforcement of aspects of it that concern computing. The current body set up by MST under the Minister of State to fashion out a suitable IT policy for Nigeria should be assisted in every way possible to complete the assignment in shortest possible time;

7. A major source of power for computers is electricity. Regular supply of electricity should be ensured in all parts of the country to enhance computing culture. Telephones (wired, mobile and wireless) are required for connectivity and computer communication. Government should evolve economic policies that would make these facilities available to the public and at affordable prices;

8. Information Technology is a universal tool. Non-producer nations acquire it to provide solutions to their problems without really having conducive environment for optimal application of the technology. For example, lack of consistent data could prevent the provision of correct solutions to problems. This partially explains why some

highly placed people acquire computers as a status symbol in Nigeria. I wish to recommend here that government should assist in establishing interlinkage programmes between local institutions (including public enterprises and educational institutions) and similar institutions in developed countries where this technology is optimally used. This will enable local institutions share expertise with foreign institutions, prepare adequately for the use of the technology and hence be in position to improve on their performance and standards;

9. As a matter of urgency, and in preparation for information technological take-off, the Federal Government of Nigeria should set up, under the Presidency, a Research and Development Centre for Information Technology (R & DC for IT). At this centre, adequate facilities for research and development of IT products should be provided. Competent researchers from any part of the world including Nigeria should be employed and tasked to contribute meaningfully to the development of Nigeria in the field of Information Technology.

10. Finally, Mr. Vice – Chancellor, let me come back to base. It is not at all palatable to remark that most important computerisable aspects of the University of Lagos operations are not computerized in this information age – staff control, resource control, examination processing, admissions, etc. are still done manually. Although, we have operations where computers are used to aid the manual processing via some sort of computerized transaction processing, this is not enough.

However, one is very much aware of the NUC package, NUMIS, that has been imposed on the Nigerian Universities in the name of management information system. For all intents and purposes, NUMIS has not made any serious impact since it was introduced over 5 years ago on the administration of the University of Lagos especially in the areas of students' records processing and financial management among others. There might

be many reasons for this inefficientiveness among which may be resistance to change of procedure in some quarters. These remarks are just observations, they are not an indictment on anybody or group of people associated with the development or use of the package.

However, I am totally convinced that with the autonomy granted to the University system, a locally produced MIS that would consider the specifics of our University's structure and mode of operations would impact more on the operations of the university. I, therefore, recommend that the University of Lagos should set up immediately an in-house committee to study and design a suitable MIS for the university. NUMIS may be used as a basis for the study. This process will take care of all appropriate operations in the University for computerization.

Furthermore, our University should embark on building a fibre-optic network to serve as backbone for a university-wide intranet so that staff and students can communicate with each other on computers. The Intranet would be connected to the Internet through a firewall. As this project is both time and money consuming, a quick, less expensive but less durable alternative can be considered as a temporary measure. The current University telephone network – PABX Hicom300E installed under my supervision as the Chairman of the Telephone Committee which has capability for voice and data communication can be hooked onto the internet via a suitable transmission equipment. This would enable the University community have access to the Internet soonest. I am happy to learn that something is currently being done along this line by the current administration.

ACKNOWLEDGEMENT

Mr. Vice-Chancellor, Sir, Distinguished Ladies and Gentlemen, I would like to recognise at this juncture some of the notable people who have contributed to making me what I am today. If I do not mention your name specifically, please forgive me, limited time and space given to me have not allowed me to mention every notable person.*

To start with, I wish to give all the honour and glory to God through our Lord Jesus Christ for sparing my life and for giving me good health to witness this wonderful day and for granting me this rare opportunity to give the Inaugural Lecture. Thank you, JESUS.

U. L. ARCHIVE

My foremost thanks go to my late parents, Mr John Olosu Ayeni and Madam Felicia Oredola Ayeni nee Alabi Bobade, the Sapetu Okitiko of Ikere Ekiti. When I reflect on my early days and consider my parents' struggles financially and spiritually to educate me (I was a little bit stubborn) I give thanks to God for their lives and for giving them courage and ability to see me through to a level I was able to fend for myself before they joined their ancestors in the 1970s. May their gentle souls continue to rest at the bosom of Jesus Christ. Amen. I also want to use this opportunity to express my sincere appreciation to my sisters, brothers and well-wishers for their unflinching support over the years.

One person I believe God used to drag me into computing is the first Professor of Computer Science in Nigeria, late Professor Olasupo J. Fagbemi. I was a survey assistant on a study leave from the Ministry of Lands and Housing, Western State of Nigeria to study Mathematics at the University of Lagos in 1968. Professor Fagbemi was the Head of Department of Mathematics then. Professor Fagbemi got interested in me when he observed that I was doing very well in the computer courses as well as other courses offered in the Department at that time. He lured me to stay on for a year to do Postgraduate Diploma in Computer Science after my first degree and to join the staff of the Institute of Computer Sciences which had just been created. He was the Dean of the Institute. Thus, I jettisoned my contract with the Ministry of Lands and Housing. I have no regrets whatsoever for taking this decision and for taking up academics as a profession. I thank God for his life. Professor Fagbemi, a brilliant scholar, an eloquent speaker, a committed academician and a die hard man for excellence died in a motor accident on Lagos-Ibadan Express Way (then under construction) shortly after he was retired from the service of the University of Lagos by the Military Administration of Muritala/Obasanjo in 1975. May his gentle soul rest in perfect peace. Amen.

I must thank brother 'Jisefinni Fasuan, a tailor by profession. God used him to take care of me for over six months when I was a job seeker after my Secondary School Certificate Examination in November/December

UNIVERSITY OF LAGOS
LIBRARY

1964. Thank you, Sir. May God always meet you at your point of need. Amen. In the same breath, I want to thank my good friend of over 35 years, Pastor Gabriel Adekunle Akinyede. I met Kunle at Federal Survey School in July 1965. We completed our Basic Surveying Examination at Oyo in May/June 1966. Both of us were posted to Akure Divisional branch of the Ministry of Lands and Housing, Western State. He arrived at Akure before me, got accommodated and settled. He accommodated me for a month before I secured my own place. Since then, our family friendship had grown from strength to strength. May God continue to uphold you and your family. Amen.

My special thanks go to Professor Laz Ekwueme for encouraging me to go ahead with my chosen topic. Although I put the topic to him as "Computing as a culture", he assured me it was a worthwhile topic. Thank you, Sir. My thanks go to my good friend, Professor Kola Ogundowole for educating me philosophically on the meaning of "culture". Thank you.

I am greatly indebted to Professor Mike Magazine, my Ph.D. supervisor at the University of Waterloo. He is the main architect of my research capability. He is unavoidably absent here today but he sent his good wishes from United States where he presently resides. Thank you, Sir, in absentia.

My thanks also go to the institutions that provide the facilities that mould my character and knowledge right from the primary school – St. John's Anglican Primary School, Ikere Ekiti, Ise-Emure Grammar School, Ise-Emure Ekiti; Federal Survey School, Okene; School of Surveying, Oyo, University of Lagos; University of Waterloo; CISM, Udine; ICTP, Trieste and C.R.A.I., Rende, Italy. I thank all the good people I interacted with in these institutions at one point or another.

I also want to thank my fellow colleagues in the Department of Computer Sciences, my students over the years and the non-academic members of the Department. I thank, in particular, members of my research group, especially Dr. Charles O. Uwadia and Dr. Dele Longe. I also want to use this opportunity to express my sincere appreciation to members of the Council of the Computer Professionals (Registration Council) of Nigeria (CPN), Computer Association of Nigeria (COAN), Information Technology (industry) Association of Nigeria (ITAN) and others not specifically mentioned but whose members are here present. May the discipline of computing continue to grow from strength to strength. Thank you all.

I must not fail to thank the crew at the Centre for Educational Technology (CET) who packaged this lecture in a most professional manner and my able assistant, Alaba Apaguntan, who wordprocessed the original script. Thank you all.

Finally, I wish to give glory to the Almighty God for giving me the pretty Bosede Alice nee Omotoso as a wife, a loving mother, a help-mate, a virtuous woman, "a jewel of inestimable value" to quote the sage, Chief Obafemi Awolowo. She is very accommodating, trusting and hardworking. She is presently the Local Administrator of Ifako/Ijaye Local Education District of Lagos State Ministry of Education. Thank you for coping with my character and for your moral support always. Special thanks also go to my children – Toyosi, who is presently studying at the University of Ottawa, Canada; Ayodeji, a graduate of Computer Science, University of Lagos. He assisted me along with Mr. Jide Ogundipe, a former PGD student in my department in retrieving information from the internet in course of preparation of this lecture; Olutosin, the baby of the family, is a level 300 student in the Department of Business Administration, University of Lagos. Olutosin, equally pretty as her mother, has been very caring. She wants to know how I feed when I travel to our country home in Ikere Ekiti. She wants to know why I work so hard always. She wants to know more, she cares very much. Children, you are a source of joy to me, I love you all. May the Almighty God care for you always. Amen.

Now unto the King Eternal, Immortal, Invisible, the only wise God, be honour and glory forever and ever. Amen (1 Tim. 1¹⁷)

Mr. Vice-Chancellor, Sir, Distinguished audience, thank you all for your attention and God bless.

UNIVERSITY OF LAGOS
LIBRARY

REFERENCES

VON NEUMANN AND MORGENSTERN (1947) : Von Neumann, J. and O. Morgenstern, *THEORY OF GAMES AND ECONOMIC BEHAVIOUR*. Princeton University Press, 1947.

KIEFER (1953) : Kiefer, J., Sequential minimax search for a maximum. *Proc. Amer. Math. Soc.*, vol.4, pp 502-505, 1953.

DANTZIG (1963) : Dantzig, G.B., *LINEAR PROGRAMMING AND EXTENSIONS*, Princeton University Press, Princeton, W.J. 1963.

BOHM (1966) : Bohm, C. and Jacopini, G., Flow Diagrams, Turing Machines, and Languages with only Two Formation Rules, *Comm. of the ACM*, 9, 5 (May), 366-371

FINE (1966) : Fine, T., Optimum search for the location of the maximum of a unimodal function, *IEEE Trans. Inform. Theory*, vol. IT-12, pp 103-111, Apr. 1966.

DIJKSTRA (1968) : Dijkstra, E.W., Goto statement considered Harmful, *Communications of the ACM*, vol.11, 3 (march), 147-148, 1968.

HAMMING (1968) : Hamming, R.W., One Man's View of Computer Science, *Journal of the ACM*, vol 16, no. 1, January 1969, pp. 3-12.

BAUER (1972) : Bauer, F.L., and Wossner, H., he Plankalkul of Konrad Zuse : A forerunner of Today's Programming Languages, *Comm. of the ACM* 15, 678-685.

SHAPIRO (1974a) : Shapiro, R.D., Optimal univariate search with unequal block sizes under minimax and stochastic criteria, Ph.D. dissertation, Stanford University, Pal. Alto, CA, 1974.

SHAPIRO (1974b) : Shapiro, R.D. and D.J. Wilde, Variable block search strategies with prior information, Dept of O.R., Tech. Rep. 74-17, Stanford University. 1974.

WAGNER (1975) : Wagner, H.M., *PRINCIPLES OF OPERATIONS RESEARCH*, Prentice Hall, Incor., 1975.

KERNIGHAN (1976) : Kernighan, B.W. and Plauger, P.J., *SOFTWARE TOOLS*, McGrawhill Book Company, 1976.

AYENI (1977) : Ayeni, J.O.A., search with prior information, Ph.D. dissertation, University of Waterloo, Waterloo, ON, Canada, 1977.

ODELEYE (1977) : Odeleye, A.O. Ayo, *A POPULAR YORUBA GAME*, Oxford University Press, Ibadan, 1977.

AYENI and MAGAZINE (1979) : Ayeni, J.O.A. and Magazine, M.J., A Heuristic Algorithm for single-variable sequential search with Prior Information, *IEEE Trans. on Sys., Man, and Cybernetics*, vol. SMC-9, no.4, Apr. 1979 pp. 246-250.

AYENI (1980) : Ayeni, J.O.A., Automatic Detection of Loop in Stepping – Stone Method of Solving The Transportation Problem, *15th Annual Conf. In Stat., Comp. Sc., O.R. and Math.* Vol. 4, O.R. Cairo University, Dec., 1980 pp. 1-24

SPRAGUE (1980) : Sprague, R.H. Jnr., A framework for the Development of Decision Systems, *MIS Quarterly* 4, no.4, 1980 pp. 1-26.

AYENI (1982) : Ayeni J.O.A., Blanks Removal from strings by a Lexical Analyzer, *Journal of Computer Science and its Applications*, vol.2, no.1, 1982, pp. 47-52.

AYENI AND MAGAZINE (1982) : Ayeni, J.O.A. and Magazine, M.J., On Optimal Stopping Rules in Sequential Seraches, *Journal of Computer Science and its Applications*, vol. 1, no.1, pp. 20-28, February, 1982.

CATELL (1982) : Cattell, R.G.G., Formalisation and automatic derivation of code generators, UMI Research Press, Ann Arbor, Michigan, 1982.

GRAHAM (1982) : Graham, S.L., Henry, R.R. and Schulman, R.A., An experiment in table-driven code generation, *SIGPLAN Notices*, 17, 6, pp. 32-43, June, 1982.

Aho (1983) : Aho, A.V., Hopcroft, T.E. and Ullman J.D., *DATA STRUCTURES AND ALGORITHMS*, Addison Wesley, 1983.

AYENI (1983) : Ayeni J.O.A., Mathematical Programming : A case study, *Nigerian Journal of Management Studies*, vol. 2, no1, 1983, pp. 100-111.

OGUNLEYE (1999) : Ogunleye, A.O., *SCIENCE EDUCATION IN NIGERIA*, Sunshine Int Pub. Ltd, 1999.

LONGE (1983) : Longe, O., Ifa Divination and Computer Science, Inaugural Lecture, University of Ibadan, Ibadan, 1983.

HENRY (1984) : Henry, R.R., Graham-Glanville code generators. Ph.D. thesis, Computer Science Division, EECS, University of California, Berkeley, CA, May, 1984.

TURKLE (1984) : Turkle S., *THE SECOND SELF : COMPUTERS AND THE HUMAN SPIRIT*, Simon and Schuster, New York, 1984.

Aho (1985) : Aho, A.V. AND Ganapathi, M., Efficient Tree Pattern Matching : An aid to code generation, *ACM Symposium on Principles of Programming Languages*, New Orleans, LA, January, 1985.

AYENI and LONGE (1985) : Ayeni, J.O.A. and Longe, H.O.D., Games People Play: Ayo, *Int. Jnr. of Game Theory*, Physicaverlag, Vienna, vol.14, issue 4, 1985, pp. 207-218.

AYENI (1988a) : Ayeni, J.O.A. A Computer Program for solving Transportation Problems, *JCSA*, vol. 5, 1, 1988, PP. 89-102

AYENI (1988b) : Ayeni, J.O.A., On Timing Traffic Lights at a Road Junction, *Modelling, Simulation & Control, C, AMSE*, vol. B, no 4, 1988, pp. 24-41.

REYNOLDS (1988) : Reynolds, G.W., *INFORMATION SYSTEMS FOR MANAGERS*, West Publishing Co., 1988.

UWADIA AND AYENI (1988) : Uwadia, C.O. and Ayeni, J.O.A., A Lag- for Preprocessor, *JCSA*, vol. 5, 1, 1988, pp. 59-68.

DENNING et.al. (1989) : Denning, P.J. (Chairman), Comer, D.E., Gries, D. et.al., Computing as a Discipline, a report, *Communications of the ACM*, vol. 32, number 1, pp. 9-23, January

AYENI (1990) : Ayeni J.O.A., A Simulation Model for Timing Road Junction Traffic Lights, *Modelling, Simulation & Control, AMSE*, vol 20, no.3, 1990, pp. 53-64.

AYENI et.al. (1990) : Ayeni, J.O.A., Longe, H.O.D., Uwadia, C.O. and Job V.O., The Game Ayo on the Computer, *NJS*, vol. 24, 1990, pp. 144-145.

UWADIA (1990) : Uwadia, C.O., Code Retargeting Using A Nucleus Set, Ph.D. Thesis, Dept. of Computer Sciences, University of Lagos Nigeria, March, 1990.

LONGE AND AYENI (1991) : Longe, H.O.D. and Ayeni, J.O.A., Ayo Game : Some Practical Difficulties and Efficiency Considerations, *JCSA*, vol.5 no.2, 1991 pp. 58-72.

AYENI (1994) : Ayeni, J.O.A., A Micro-based prototype Students' Administration System, Research Report no. CRS 88/04, University of Lagos, Akoka, 1994.

MALLCH (1994) : Mallach, E.G., *UNDERSTANDING DECISION SUPPORT SYSTEMS AND EXPERT SYSTEM*, IRWIN, 1994.

UWADIA and AYENI (1994) : Uwadia, C.O. and Ayeni, J.O.A., A methodology for obtaining the nucleus set of a group of instruction sets, *Ife Journal of Technology*, OAU, vol., no1, 1994.

AIMUFUA AND AYENI (1996) : Aimufua U.E. and Ayeni, J.O.A., A Time-Table Improvement Algorithm, *JSCA*, vol.2, 1995/96, pp. 145-154.

AYENI AND UWADIA (1996) : Ayeni, J.O.A. and Uwadia, C.O., A Micro-based Prototype Students' Administration System, *JSRD*, vol.2, 1995/96, pp. 162-171.

ZHI et.al. (1997) : Zhi, Y.C., Wang, P. and Turban, E., Management Support Systems of State-Owned Enterprises in China, *Int. Jnl. Of Information Management*, vol.17, no.4, pp. 271-285, 1997.

GI TASK-FORCE (1998) : Government of India IT Taskforce, Information Technology Action Plan, <http://it-taskforce.nic.in/it-taskforce>, 1998.

AYENI AND BAMGBOSE (1999) : Ayeni, J.O.A. and Bamgbose, A.P., A Faculty Students' Information System, *JSRD*, vol 4, 1999, pp. 137-154.

OGUNLEYE (1999) : Ogunleye, A.O., SCIENCE EDUCATION IN NIGERIA, Sunshine Int. Pub. (Nig) Ltd, 1999

COAN (2000) : THE COMPUTER ASSOCIATION OF NIGERIA (COAN), NIGERIA SOFTWARE EXHIBITION-NISE 2000, Brochure, March/April 2000.

FMST (2000a) : Federal Ministry of Science and Technology, NATIONAL REPORT ON THE MILLENIUM (Y2K) BUG, DRAFT, vol.3, Abuja, Nigeria, 2000

FMST (2000b) : Federal Ministry of Science and Technology, NATIONAL REPORT ON MILLENIUM (Y2K) BUG, vol.1, MAIN REPORT, May, 2000.

NJOKU (2000) : Njoku, C.N., Learning System Dynamics through Knowledge-base Updating, Experimental Study with Jonda, Ph.D. thesis, Department of Computer Sciences, University of Lagos, 2000.

NJOKU AND AYENI (2000) : Njoku, C.N. and Ayeni, J.O.A., A theory of Jonda, Technical Paper, Department of Computer Sciences, University of Lagos, Nigeria, 2000.

RALSTON (2000) : Ralston, A., Reilly, E.D. and Hemmendinger, D. (eds.), Encyclopedia of Computer Science, Fourth Edition. Nature Publishing Group, 2000.

THE WORLD (2000) : www.dci.gov/cia/publications/factbook/geos/ni.html 2000.

