National Engineering Conference Series Vol. 5, No. 1, pp125-131,
1998, Copyright © 1998, NEC Publications. Printed in Nigeria
 ISSN 1118-8383

PROSPECTS FOR SOFTWARE DEVELOPMENT FOR INDIGENOUS DESIGNS

SIMOLOWO O. EMMANUEL
Department of Mechanical Engineering, University of Ibadan, Ibadan.

ABSTRACT
The importance of software development in the world today cannot be over-emphasised, moreso as applied in the to the field of engineering. Software development for indigeneous engineering designs needs to be given peculiar considerations as it entails developing software from the basics (high level language) for it to handle target indegeneous design specifications.

This paper therefore in its focus starts from the central point of all computed aided engineering activities called the "Database". The database discussed entails (i) Documentation (ii) Synthesis and analysis. Tools for software development such as computer languages and hardware configurations available within the country for indegeneous software development have also been considered. Common computer language modes and their functions as well as practical techniques of packaging developed software have also been outlined. The design chart of the software for generation of Cam profiles developed in a research work (CAD of Cam profiles) based on the application of procedures discussed in this paper is shown as reference figure. Majority, research work in this area will alleviate the problem of much dependence on imported software whose applications are limited as regards local design and also not fully utilized due to the little understanding of the software flexibility and capabilities by unskilled users.

INTRODUCTION

Developing software for indegeneous designs will serve as burden bearer for many expensive, tedious, undocumented engineering designs carried out in this part of the world. It will also help to preserve documented engineering designs developed over the years in the forms of software packages. These documented designs and be re-examined for modifications at less tedious and expensive procedures.

Expansion of academic knowledge in this area of Indegeneous Software Development will be enhanced as he software produced could be used as learning and teaching aids in higher institutions of learning. The geometric database and its role has been analysed. Reasons and advantages that makes 'C' language Turbo C, C++ e.t.c.) one of the idle high level languages for software development for indegeneous designs have been considered. Outstanding computer language modes and functions of the idle high level language ncludes (i) The ANSI Cursor System Control, (ii) The Spawnl Function (iii) The Screen Print System Command. Techniques for packaging completed software for universal use have also been fully analysed.

DATABASE FOR INDEGENEOUS DESIGNS

The source or destination of data, information and series of procedures which are used in a given operation s to generate the expected result when processed is a database. When such data are used to generate eometries (shape and properties) of designs it is known as a geometric database. Geometric database for ndigenous as shown in figure 1 would comprise of two basic procedures (i) documentation (ii) Synthesis and nalysis.

Documentation

his process involves the production of drawings of indigeneous or locally designed machines or equipments 2-D, 3-D of the assembly and constituents parts and also solid representations of objects or machines esigned. This documentation process will help in the collation of various designs done locally through esearches and manufacturing processes.

2-D Images/Drawings

For 2-D images, the geometric database when processed will perform the following operations.

i production of 2-D drawings of assembly units and their constituent parts;

) transmission of geometry to other programs for manipulations, design, analysis and manufacturing processes;

(iii) transmission of geometry to be used in management activities.

3-D Images/Model

Processes carried out by the geometric database when processed for 3-D drawings will include:

(i) creation of 3-D drawings e.g. Isometric drawings from constituent parts;
(ii) generation of automatic views by applying the principles underlying autographic projections;
(iii) performing the functions of 2-D geometric database.
The 3-D models are of three types as in computer aided engineering (i) Wire frame (ii) the surface modelling
(iv) solid modelling.

The geometric database in wire framing extends the 2-D draughting approach to 3-D space. It further creates object images using lines and curves but without surfaces. The geometric database creates surfaces for the models generated in the surface modelling processes. In solid modelling, the geometric database not only generates surfaces for models created but also the internal parts that cannot be seen. The generated object image when sectioned reveals all hidden parts as in the real objects its self. All functions performed by the geometric database in wire - framing and surface modelling are also carried out by 3-D modelling data-base.

Synthesis and Analysis

The process of synthesis involves the combination of all mathematical, logical and graphical parts of the program being developed. Standard and newly applied equations are analysed for the design been developed. Numerical results are obtained while various design checks are carried out to ascertain the validity of Design. Processing of this aspects of the data-base gives numerical output of data that is used for practical manufacturing of the design been carried out. This is not shown on cam chart in figure (3) of the reference software chart. Results obtained from this aspect of database can also be transferred to other documentation aspect of the database for further processing.

TOOLS FOR INDIGENOUS SOFTWARE DEVELOPMENT

High-level Languages

The high-level language employed in computed aided design which are also applied to software development for indegeneous designs include the following:

(i) BASIC (ii) Assembly Language (iii) FORTRAN (iv) PASCAL (v) C/C** Language.

The BASIC could be used to develop softwares for designs in the areas of simple types of computer graphics. Database programs are also written in basic and manipulated. A high level language such as BASIC might run too slowly for dynamic displays in computer designs hence part of the programs could be written in assembly language or code called up from the main BASIC program.

C and Pascal Languages. These are the high-level languages that are becoming popular for complex graphics.

Idle Language for software Development

The advantages of C language that makes it an idle high level language for software development and more so for indegeneous designs are as enumerated following [3].

(i) Well structured language for software development and maintenance.
(ii) Ability to handle large and complex programs.
(iii) High speed compilation and generation programming.
(iv) Ability to perform similar operations as on the IBM in assembly language much more conveniently using C language.
(v) Portability of compiler
(vi) The Standard library concept
(vii) Powerful and varied repertoire of operations.
(viii) Elegant syntax, ready access to the hardware when needed.
(ix) Ease of optimising applications.

Hardware for Software Development

The type of simple Computer Aided Design (CAD) systems found in small firms and institutions of higher learning in this part of the world consists of a small PC based system with a supplementary memory and keyboard. However simple CAD stations could be balanced up with open architecture computing.

It is important to choose hardware that has flexibility to take advantage of the modifications in software or hardware technology. Care must be taken to ensure that the most appropriate screen, graphics cards, adaptors, and co-processors are selected for the intended software system. The colour graphics capabilities of these hardware can be enhanced using standard graphics adaptors e.g. V.G.A. (Video Graphics Array) and EGA (Enhanced Graphics Adaptor) as well as maths co-processor or extended Ram memory.

FUNCTIONS MODES AND COMMANDS

The language mode functions described in this section include those that were used to develop the software for designing and generating profiles of various Cam types, the chart of which is shown in Figures (3). The language modes used for software development generally and which can also be applied to indegeneous designs based on C/C^{++} high level language apart from the text and graphics includes the ANSI Cursor system control. The Spawnl command and systems command for printing screens.

The ANSI Cursor System

This Cursor System control is used in the reference software (CAD of Cams). The system Cursor control is to directly control the position of Cursor on the screen. However, since this system does not exist on the Read Only Memory (ROM) of the computer, software algorithms have to be written in connection with the existing ANSI files located in the operating systems as to make cursor control operative. The usage of the system control is employed for selecting the various options on the menu table on every screen display. To use the ANSI system steps are taken to install the system driver in the operating systems [3].

The Spawnl Functions

This command makes the compilation of large programs of thousands of lines possible by a method called child processing. The methods involves (i) breaking the large program into smaller program modules (child programs) (ii) compiling the program modules separately (iii) Combining the execution files of already compiled programs. (iv) Finally the spawnl command returns control to main program after running the child processes.

The System Command

The printing of screens using the high level C/C^{++} language is not possible without the use of the systems command. The systems command once incorporated into the main program enables the programmer produce hard copies of various designs as in other universal software.

SOFTWARE PACKAGING TECHNIQUES

The creation of software packages that can be used universally outside the environment of their development is an area that has lacked adequate understanding in the whole process of software development, moreso in this part of the world. Outlined here in below are tested techniques in stages (see figure 2) for creating stand-alone software as applied to indegeneous designs.

Stage 1: Algorithm Development

This stage covers all processes of developing software algorithms described in section 1-3. Programs larger than about 4,000 lines of algorithm have to be broken into smaller seperate programs called program modules. The sample program of chart shown in figures 3 was broken into four program modules (Cam design 1-4).

Stage 2: Compilation of program modules

All program modules are at this stage developed into complete algorithm and made to become seperate programs that can be compiled. The compilation of the program modules are carried out using the high level compiler and their seperate execution files are generated shown as Module 1. Exec. to Module 4. Exec. in figure 2.

Stage 3: Combination of program modules

Using the spawnl command in C^{++} language all program modules are combined to form one program called the main program. Recompilation of the main program is then carried out and the overall execution file for the main program is then obtained.

Stage 4: Temporary Program Packaging

A subdirectory is created under the main root directory containing the C^{++} compiler. All program module

execution files as well as that of the main program are copied into the new subdirectory. Since the programs are still within the environment of the compiler, the package is tested by typing just the name of the main program at the subdirectory prompt. One entered the software operates as a stand-alone package but in the compiler's environment.

Stage 5: Final Program Packaging

At this stage the software package developed is made free of the compiler environment. To achieve this (i) The whole execution programs (modules and main programs) are copied into the final external storage units e.g. diskettes, disks, e.t.c. (ii) All the files needed for compilation are extracted from the main compiler in the root directory of the environment of software development. (iii) Extracted files (see fig. 2) from compiler are also copied into the initial external storage unit in which the program execution files were copied. (iv) The package can then be tested by typing in a name as pertaining to the main program once loaded on any system outside design environment.

CONCLUSION

Prospects for software development of Indegeneous design are high as all the tools (hardware, software and high level language texts) are now available in most developing countries. A notable distinction has been made between software development for indegeneous designs which entails building algorithms from high level languages such C/C**, Pascal e.t.c. and the use of already developed packages for design procedures.

The gap existing in the knowledge of development and usage of software between the advanced and developing countries can be reduced and eventually closed up with the introduction of a realistic foundational base for developing software packages for indegeneous designs in this part of the world. The large amount of money spent and lost on imported software and industrial setbacks as a result of lack of proper use and inadeaptability of such imported software to indegeneous and local designs can be greatly reduced with the advent of indegeneously designed software packages.

REFERENCES

1. *Davies, B. Robothan. A. Yarwood A. "Computer Aided Drawing and Design".*
2. *Charles, S. "Teach Yourself C".*
3. *Robert L. "Turbo C Programming for the PC".*
4. *Paul R. Dean C., "Drafting in Computer Age"*

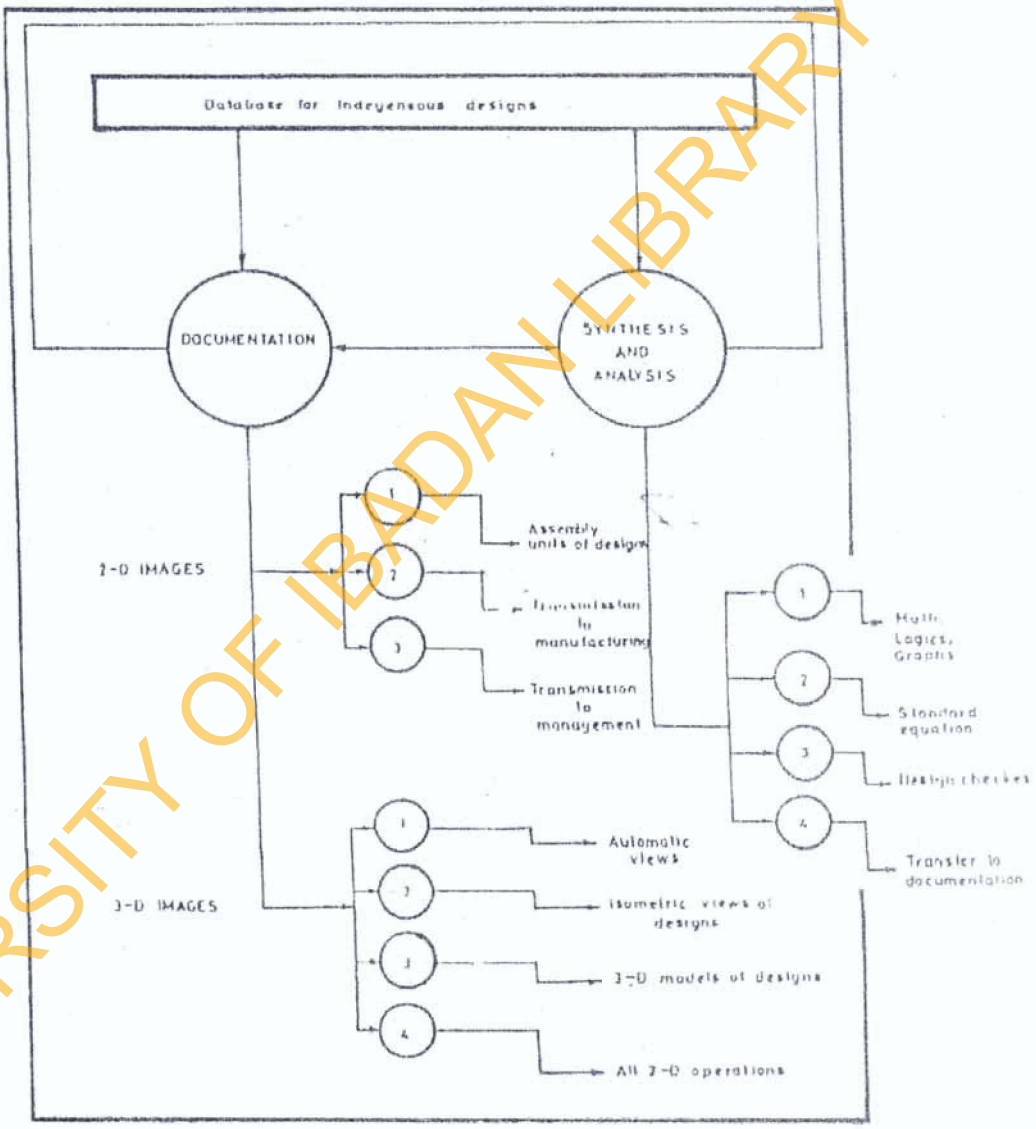

Fig. 1. Functions of the database for Indegen designs.

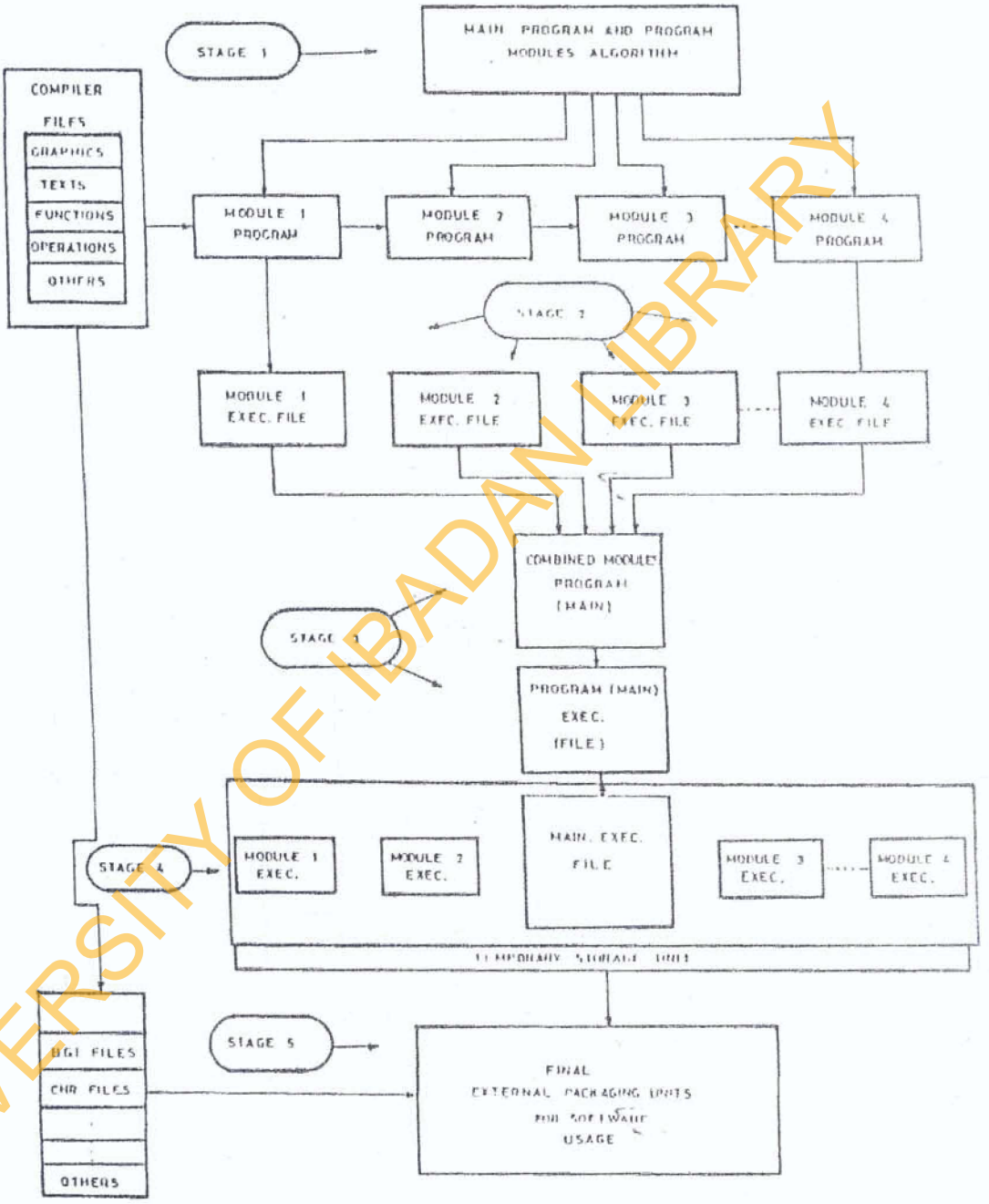

Fig. 2. Tested method for software packaging steps and techniques based on software developed (read of com's)

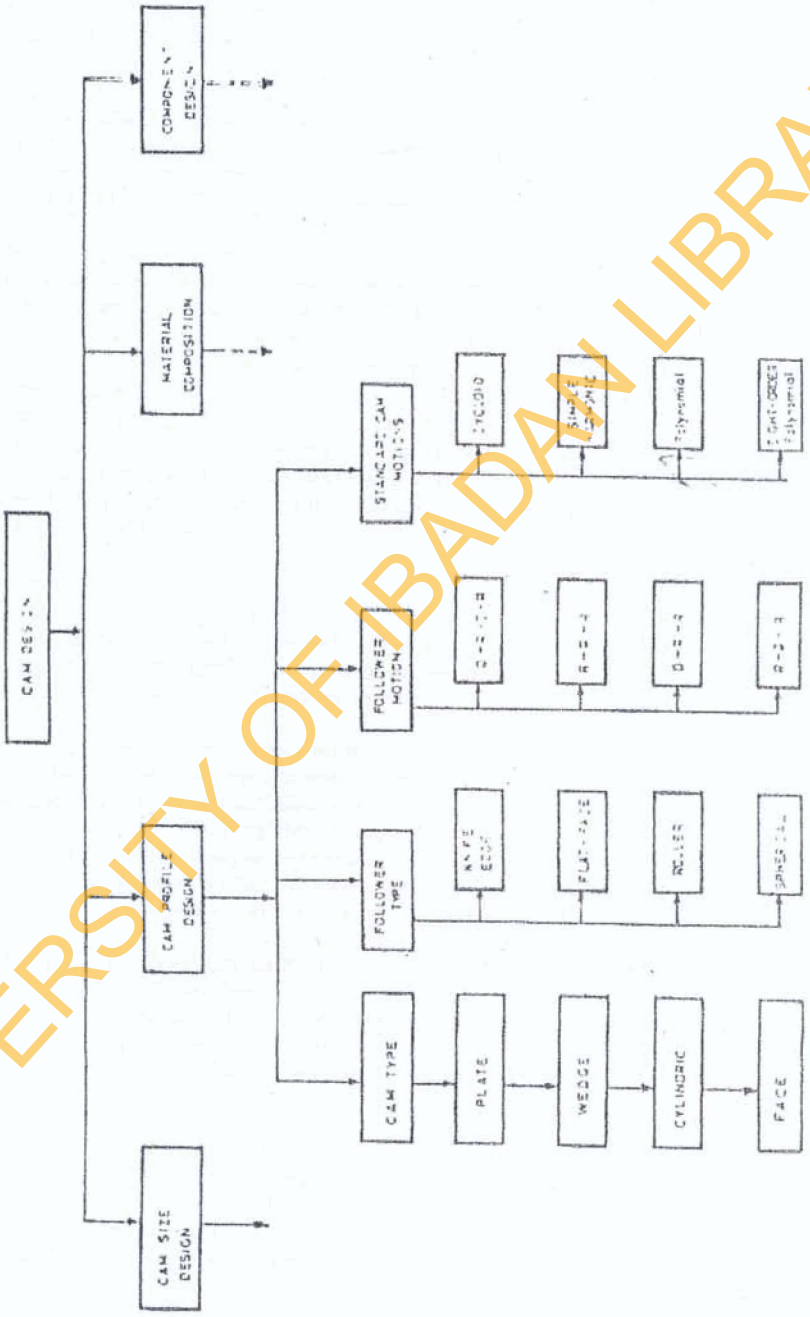

Figure 3: Flow chart of overall design options of developed software for CAD of Cams.