DYSBASE : DYNAMIC SIMULATION WITH DATABASE
(A Software for System Dynamics Modelling)

M.V.S.L.NARAYANA

AND

SUSHIL

CENTRE FOR MANAGEMENT STUDIES
INDIAN INSTITUTE OF TECHNOLOGY, DELHI
HAUZ KHAS, NEW DELHI - 110 016 (INDIA)

ABSTRACT

System Dynamics is one the methodologies for behavioural research. Some of the important steps for the behavioural analysis under system dynamics framework are system conceptualisation, model fabrication, model simulation and finally the policy analysis. Out of these steps model simulation usually involves massive computation and hence it may necessitate the usage of computers. When it is a question of the usage of computers there is a need of a language for computer oriented model preparation and simulation. To cater to this need a software viz. DYSBASE (DYNAMIC SIMULATION WITH DATA BASE) has been developed. This paper chalks out an overview of DYSBASE. The language syntax and the operational procedure of DYSBASE software is described in the DYSBASE manual available with the authors.

BACKGROUND OF SYSTEM DYNAMICS METHODOLOGY

System dynamics is a methodology for understanding complex systems. The System Dynamics (SD) methodology puts the structure, policies, information, experience and various other flows of an organisation into a mathematical model that can be simulated on a computer to understand the dynamic behaviour of the concerned organisation. System Dynamics is based on the following premises (Forrester,1969).
(a) The dynamic behaviour of any organisation is governed by it´s structure and policies.
(b) An organisation can be better viewed in terms of it´s underlying flows rather than in terms of it´s functions. The flow may be the flow of information, money, people, capital equipment, orders, etc.

SOFTWARE TOOLS FOR SYSTEM DYNAMICS MODELLING

Apart from the hardware there is a strong need of good software tools to represent the System Dynamics models on a computer in a machine understandable form and to simulate them over a given period of time. Some of the popular software tools are : (a) DYNAMO, (b) STELLA, (c) DYSMAP and (d) DYMOSIM (India). Of these DYMOSIM was developed by I.I.T., Kharagpur. It needs the user to know FORTRAN language thoroughly. STELLA has an additional facility of drawing the flow diagrams for the SD model which is quite appreciable. An effort has been made to develope a new SD package viz. DYSBASE which is easy-to-use and provides an infrastructure for developing integrated models using other modelling techniques in conjunction with SD.

DYSBASE - AN OVERVIEW

DYSBASE is a PC based user friendly and menu driven software package for System Dynamics modelling. The salient features of DYSBASE are as follows.

(a) DYSBASE provides a Relational Database environment also. This enables the user to maintain a large database of simulated results for future use.
(b) DYSBASE is a completely menu-driven user friendly package
(c) It has it´s own Editor (written in PASCAL) to enter and edit the programs written in DYSBASE language to create executable module of the program.
(d) DYSBASE has a graphics module (written in C language) (Johnson,1989;Kernighan,1977). It helps in drawing the XY and BAR charts with the simulated results of an SD model.
(e) It has a Table Maintenance Module to create, modify or delete TABLES used in conventional SD modelling.
(f) The database files generated by DYSBASE can be easily transported to some of the popular IBM PC softwares like LOTUS 1-2-3 (an electronic spreadsheet), SYMPHONY, SUPERCALC (an electronic spreadsheet) etc.
(g) We can run programs written in languages like ASSEMBLY, ´C´ , FORTRAN and CLIPPER (a relational Database) directly from DYSBASE without any compatibility problems. This is possible through the Extended Module of DYSBASE.
(h) The programming language provided by DYSBASE consists of only 6 Statements and a few functions. Hence SD modelling using DYSBASE language is easy to learn.
(i) In BAR graph there is a provision of having a halt after plotting the values of the variables at each interval. The values of the variables for the next time interval will be plotted only when the user signals the system by pressing any key of the keyboard.
(j) It is available on PCs and hence it is within the reach of most of the users.

DYSBASE SOFTWARE STRUCTURE

When DYSBASE is invoked the Main Menu of the software appears. The "Main Menu" consists of the following options. Most of the screen forms of DYSBASE have been designed using CLIPPER (Autumn,1987).

(a) TABLE (b) EDIT (c) COMPILE (d) RUN (e) RESULT (f) GRAPH

A brief description of each of these options is as follows.

(a) TABLE : This option is used to create, modify or delete tables used in the system dynamics model.
(b) EDIT : This option from the main menu invokes the inbuilt text editor. This text editor is a very simple editor equipped with necessary commands to edit a given program or to write a new program into a file.
(c) COMPILE : This option is chosen to compile a given program written in DYSBASE. Before compiling any program this module checks the presence of all the necessary system files for the compilation of a program. If DYSBASE is not properly installed then it displays an error message indicating missing file names.
(d) RUN : This option is used to run a compiled program. Before executing this option the user must ensure that he has compiled the program selecting the "Compile" option from the main menu.
(e) RESULT : This option of the main menu is meant for entering the legends of the variables of the result file of a particular SD model. Also it can be used to have a hard copy of the values of the variables stored in a particular result file.
(f) GRAPH : This option is used for drawing XY or BAR graph with the data of the databâse built by running DYSBASE model in DYSBASE.

HIERARCHICAL STRUCTURE CHART OF DYSBASE

DYSBASE is a structured modular software. Each module consists of several programs. The methodology adopted is structured programming technique. The hierarchical structure chart of the software is shown in Fig. 1. It shows the calling-called relationship amongst the different programs of the software. All the programs of DYSBASE are classified into 5 levels.

Level - 1 : It consists of DYS.PRG program. It displays the main menu of DYSBASE.

Level - 2 : It consists of the following programs.

(a) INST_CHK : It checks whether DYSBASE has been properly installed or not.
(b) TBLCRT : It displays a set of options (like addition, deletion, modification, etc.) to maintain the tables to be used in an SD model.

HIERARCHICAL STRUCTURE CHART
OF DYSBASE SOFTWARE

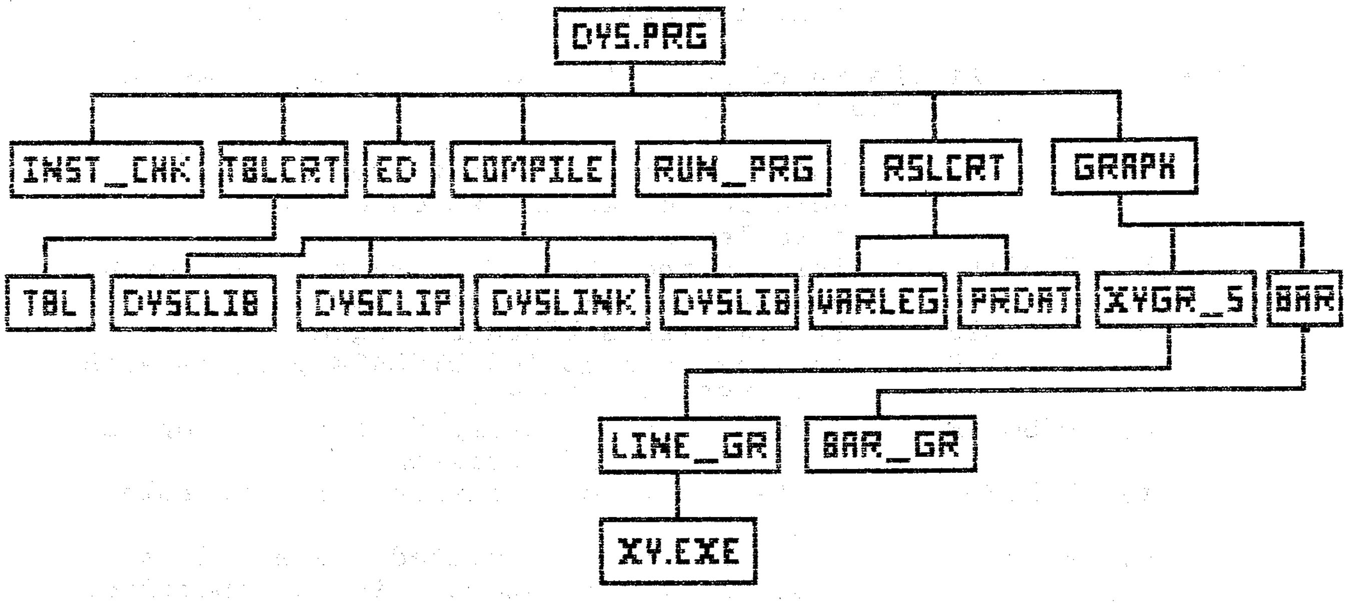

Fig. 1

(c) ED.EXE : It is the inbuilt editor written in PASCAL. This is complete structured software in itself with more than 25 programs.

(d) COMPILE : It displays a window asking the user for the DYSBASE program file to be compiled and linked.

(e) RUN_PRG : It runs the compiled and linked DYSBASE program.

(f) RSLCRT : It displays a menu providing the user with an option creating variable legends and printing the simulated data. The variable legends are in the graphs for identification purpose.

(g) GRAPH : It displays the menu consisting of the graphics option like XY, BAR, etc.

Level - 3 : At the third level the following programs are there.

(a) TBL : This program is meant for addition, deletion or modification of the data in a table file.

(b) DYSLIB : This is the DYSBASE library file having all the DYSBASE functions.

(c) DYSCLIP : It compiles a DYSBASE program.

(d) DYSLINK : It links a compiled DYSBASE program with the DYSBASE library.

(e) DYSCLIB : It is another library file needed by a compiled DYSBASE program.

(f) VARLEG : It is meant for creation of variable legends.

(g) PRDAT : It prints the simulated data of a DYSBASE model stored in a specified database file.

(h) XYGR_S : It displays a window prompting the user to give all the parameters necessary to draw an XY graph.

(i) BAR : This program displays an interactive window to accept the parameters necessary to draw a bar graph.

Level - 4 : This level consists of two files.

(a) LINE_GR : It is a data format converter program. It converts the data of a relational database into a standard data file (text file) suitable for plotting an XY graph.

(b) BAR_GR : It is called by the program viz. BAR to draw the bar chart.

Level - 5 : The fifth level has a single program described below.

(a) XY.EXE : It the executable version of the program

XY.C written in C language called by LINE_GR program to draw the xy graph. To draw an xy graph it calls some even lower level functions as given below.

(i) VMODE(x) : It changes the VDU mode from text to graphics and vice versa.
(ii) WRDOT(X,Y) : It writes a pixel on the screen.

(b) PRT_GR : This program employs the printer pin programming technique to print the graph.

DYSBASE LANGUAGE SYNTAX

DYSBASE provides an easy-to-learn language for SD modelling. The syntax of DYSBASE programming language is as given below.

Variables and constants in DYSBASE

(a) Variables : The variables can be named using not more than 1Ø characters. The first character of a variable name should not be a digit. The permitted character set for variable names is as follows.
Ø through 9, Underscore, A to Z and a to z.
Example : CONST, OR_JK etc.

(b) Constants : All integer or real numbers are constants in DYSBASE. Example : 1,23,1Ø.3,-15.23,-16 etc.

Arithmatic Operators in DYSBASE

Following are the arithmatic operators in DYSBASE.

+ (Addition), - (Multiplication), * (Multiplication), / (Division), ** (Exponentiation)

Variable Notation

It is better to use the variables in DYSBASE with time scripts indicating their place in time. K denotes the present, J is the point in time just past and L the point in time in the immediate future. The symbol DT is used to represent the length of time between J and K or K and L. The variables representing levels may have J or K time scripts. Whereas the rates can have time scripts JK or KL to represent the rate variables of past period and future period respectively. There should be an underscore between the variable name and it´s time scripts symbol i.e. J,K,JK or KL.
Example : INV_K = INV_J + DT * (ORDER_JK - SHPMTS_JK)

```
     where INV_K = Value of inventory now.
           INV_J = Value of inventory a time interval ago.
              DT = Length of the time interval.
        ORDER_JK = Order received over the time interval JK.
```

SHPMTS_JK = Shipments over the time interval JK.

DYABASE Statements

Following are the statements of DYSBASE language.

(a) PUBLIC DT : This statement is required to set the time interval. It should be the first statement of any DYSBASE model. Here DT is the variable representing the time interval and we assign any desired value to it explicitly.

```
Example : PUBLIC DT
          DT = Ø.25
```

(b) SELECT statement : This statement selects an area in the main memory to open a file. Before opening a database file we must select an area for the file.
(c) USE statement : It is used to open a file.
(d) DO WHILE Statement : This statement is used to construct an iterative loop. It executes a set of statements iteratively till a given logical condition remains satisfied.
(e) COMMENT LINES : DYSBASE has also a provision to write comment lines any where within a DYSBASE program. These comment lines increase the readability and understandability of the program. Each comment line should be preceeded by a star (*) symbol.
(f) DO WITH statement : This statement is used to run a dBASE-III(plus) or CLIPPER program from a DYSBASE program with some parameters. It is noteworthy here that the dbase-III or CLIPPER programs called by DO statement need not be compiled individually.
(b) IF statement : This statement is used for conditional execution of a given set of statements.

DYSBASE Functions

Following are the functions supported by DYSBASE. These functions are very useful in preparing SD models.

(a) STEP() : This function, as the name itself says, is used to change a quantity abruptly at some point in time.
(b) TABLE() : It takes one quantity as an input and finds out the corresponding value of another quantity, which is having a non-linear relationship with the first quantity shown by the values φf a given table.
(c) SMOOTH() : This function finds it´s application in averaging a quantity over a given period of time.
(d) DELAY1() : This is a function for 1st order delay.
(e) DELAY3() : This a function for third order delay similar to DELAY1().
(f) DLINF1() : This is a 1st order information delay function.
(g) DLINF3() : This is a 3rd order information delay function similar to DLINF1().
(i) SQRT() : This function returns the non-negative square root of a given quantity (X).
(j) SIN() : It returns the Sine of a given angle.

(k) COS() : It finds the cosine of a given angle.
(l) EXP() : It is the exponential function. It returns the value e^x.
(m) MAX() : It returns the maximum of two given numeric quantities.
(n) MIN() : Returns the minimum of two given quantities.
(o) CLIP() : This function returns a value A as long as X>=Y otherwise it returns B.

$$OUT = \begin{cases} A & \text{if } X >= Y \\ B & \text{if } X < Y \end{cases}$$

(p) SWITCH() : This function returns a specified value A if a given quantity (X) equals zero else returns some other specified value B.
(q) RAMP() : It is a continuously growing or declining function of Time.

GENERAL STRCTURE OF A DYSBASE PROGRAM

A DYSBASE program has mainly 3 divisions.

(i) Initialization Division : All the variables (levels, rates, auxiliaries or others) being used in the program are initialized here.
(ii) File Division : In this division we select different areas in the main memory of the computer using SELECT statement to open different database files and tables to be used in the DYSBASE program.
(iii) Procedure Division : This section actually describes the complete procedure of the DYSBASE program in terms of some equations of the SD model.

```
Example :  * Initialization Division
                PUBLIC DT
                PP_J = 1ØØ
                RR_JK = Ø
                IR_JK = Ø
                ASL = 5
                SH = 4Ø
                ST = 3
                PC = 2ØØØ
                DT = 1
                CT = 1
                ECTS_K = Ø
                OFFSET = Ø
                *  File Division
                SELE 1
                USE PPT
                * Procedure Division
                DO WHILE CT <= 5Ø
                  PP_K = PP_J + DT * (IR_JK - RR_JK)
                   CR_K = PP_K / PC
                   SELE 1
```

```
          TABLE(CR_K,"ECTS_K")
          ATS_K = ASL * ECTS_K
          RR_KL = PP_K / ATS_K
          NIR = PP_K / ATS_K
          STEP(SH,ST,CT,"OFFSET")
          IR_KL = NIR + OFFSET
          RESULT("PPM",2,PP_K,RR_KL,Ø,Ø,Ø,Ø,CT)
          CT = CT + 1
       ENDDO
```

GUIDELINES FOR WRITING A FLAWLESS DYSBASE PROGRAM

To write a flawless DYSBASE program the following points must be kept in mind.

(a) First prepare the arrow-diagram for the given problem and then prepare the flow diagram using conventional SD symbols (Goodman,1982;Richardson,1981).
(b) Write down the equations using DYSBASE language syntax henceforth known as DYSBASE statements.
(c) The following are the points to be kept in mind for easy sequencing of DYSBASE statements.

(i) If an equation - A is dependent on the other equation-B then equation-A is preceded by equation - B. An equation is said to be dependent on another equation if it consists of a variable which is evaluated by the other equation.

Example : Let us take the following three DYSBASE equations.

```
  TABLE(POLR_K "PAT_K")        ---------- (A)
  POLR_K = POL_K / POLS        ---------- (B)
  POLAR_KL = POL_K /PAT_K      ---------- (C)
```

Here we find that the first equation - A which is essentially a TABLE function call needs the POLR_K as an input value and then it evaluates PAT_K.
Equation - B evaluates POL_K. Hence equation - B must be written prior to equation - A. Now equation - C computes POLAR_KL and for this purpose it needs both POL_K and PAT_K. Since PAT_K is evaluated by equation - A, it must be written before equation - C. However, here equation - B and equation - C are independent equations because none of them needs a variable evaluated by the other. Hence equation - B and equation - C may be in any sequence amongst themselves but both of them must be written after equation - A. Thus the only possible sequence of the three equations is :

```
     POLR_K = POL_K / POLS
     TABLE(POLR_K,"PAT_K")
     POLAR_KL = POL_K / PAT_K
```

(ii) Two independent equations may be written in any order.
(d) The last statement of the procedure division must be the

result function call. All the procedure statements including the result() function call must be kept within a DO WHILE loop. Example :

```
DO WHILE C <= LIMIT
 INV_K = INV_J + (OR_KL-SR_KL)
 DISCR = DINV - INV_K
 OR_KL = FOW*DISCR
 STEP(20,4,C,"SR_KL")
 RESULT("AA1",3,INV_K,OR_KL,SR_KL,0,0,0,C)
 C = C + 1
ENDDO
```

CONCLUSION

DYSBASE is a useful software for SD modelling characterised by some important features like user-friendliness, compatibility with easily affordable IBM personal computers and easy-to-learn. It provides a low cost solution to SD modelling. Also the extended module of DYSBASE supports the interface with programs written in C, PASCAL, BASIC, FORTRAN, CLIPPER or FOXBASE language.

ACKNOWLEDGEMENT

The authors wish to acknowledge the contribution of Dr. A. Kanda, Department of Mechanical Engineering, I.I.T. Delhi in the development of the software.

REFERENCES

1. Autumn 87, CLIPPER Manual.
2. Coyle R.G. (1977), Management Systems Dynamics, John Wiley & Sons : London.
3. dBASE-III (Plus), SYBAX Series Manual.
4. dBASE - III (Plus) Manual (1986), Aston Tate Corporation, USA.
5. Forrester J.W. (1969), Principles of Systems , Wright Allen Press Inc., Cambridge, Massachusetts, USA.
6. Forrester J.W. (1968), Workbook for Principles of Systems, Wright Allen Press Inc., Cambridge, Massachusetts, USA.
7. Goodman M.R. (1982), Study Notes in System Dynamics, MIT Press, Massachusetts, USA
8. Jamsa Kris A. (1989), Programming in C, Richard Dirwin Inc. Publication, Colorado, USA.
9. Kernighan Brian W. and Dennis M. Ritche (1977),C Programming language, Prentice Hall International Inc., Englewood Cliff.
10. Richardson George P. & Pugh Alexander L. (1981) , Introduction to System Dynamics Modelling with DYNAMO, The MIT Press, Cambridge, Massachusetts (USA).