SYSTEM DYNAMICS SIMULATION SOFTWARE ———DYNAMOX

Jiuqiang Han, Guoji Sun
System Engineering Institute,
Xi'an Jiaotong University, Xi'an, PRC

ABSTRACT

The purpose of this paper is to present the characteristic, functions and program structure of system dynamics simulation software ——DYNAMOX that can be used on XENIX operation system. This software is mainly composed of six parts. (1) dispatch management module, (2) parameter modify module, (3) model compiler module, (4) model run module, (5) database access module, (6) model library access module. In order to illustrate our software, we give a simple example about the organisms relationship model between deers, beasts and grass in Kaibab plateau.

INTRUDUCTION

System dynamics method and simulation technique is a very good study way that si used to plan and decision the complex systems for society、economics、management、biology、population and enviroment etc.. In system donamics method, the main tool used to study those problems is a system dynamics simulation software。This kind of simulation software was earliest developed by MIT. Late on, it is popularized to all over the world. The system donamics software called Micro ——Dynamo was introduced into china in the 1980s, because its model size is too small, it can't become a support enviroment used to scientific study, so it is replaced by another system dynamics simulation language DYNAMOC developed by authors. DYNAMOC has played important role in pushing the research and application of system dynamics method forward in china. It has been extensively used to the study on society, economics, population and environment system etc.. But, the application filed of system dynamics simulation technique is becoming more and more wide, the problem to be researched more and more complex, and the eqaution size to be solved more and more huge. Particularly, it can not meet the needs of simulating many large scale economic decision support system modles. In addition, there also exist some disadvantages, such as, limited model size, single user, data no leaving from modle and lack of data common enjoy (

share) capability etc.. Hence,in order to overcome all those shortcomings , we have developed a new system dynamics simulation software ———DYNAMOX that can be run on XENIX operation system. DYNAMOX software has successfully been used in "the decision support system about science、technology 、economic and society harmonize development in SHAAN XI province ".

THE CHARACTISTICS OF THE SOFTWARE

Many User

Data common enjoy (share): DYNAMOX can be used by many users on XENIX support enviroment, and provides the function to operate database. When many users use DYNAMOX,the data in database can be shared by each user. Data leaving from modle: In traditionary system dynamics simulation software, data can't leave from modle so that simulation result data of different modles can not be used each other. DYNAMOX can storage user modle to modle library and simulation data to database respectively. Model Size: In DYNAMOX, system dynamics simulation model size is only limited by memory space in computer because of using dynamic memory allocation technology. DYNAMOX can simulate system dynamics model with 10000 equations on personal computer.

Algorithm and Interactive

Many integration methods: DYNAMOX provides three kinds of integration rules included Euler,fourth order Runge——Kutta and changable step. These integration methods can entirely meet the needs of different simulation accuracy and speed. Many kind of equations and functions: DYNAMOX can solve all kinds of equations and functions stipulated in s tandard DYANMO langauge. In addition the independent variable of table function may be both equal and unequal interval, and provides relative table functions too. Parameter optimization: DYNAMOX can optimize five model parameters each time. After user enters the parameters to be optimized and objective function according to the syntax of DYNAMOX, the parameters can be optimized immediately. Interactive: Because of using menu technique full screen and providing large scale assistance operation informations, when user only enters a few commands, simulation process may be completed. During simulation run, user can pause the run at any time, and modify parameter and display data etc.. After simulation end, user can also modify parameter and rerun, but it is unnecessary to recompile modle.

THE STRUCTURE OF THE SOFTWARE

The structure of system dynamics simulation software program is mainly composed of ten parts, i. e. man—machine interface part, dispatch management, parameter modify, data access, model access, model compiler, model run , database and model library. The software structuer is shown in Figure. 1.

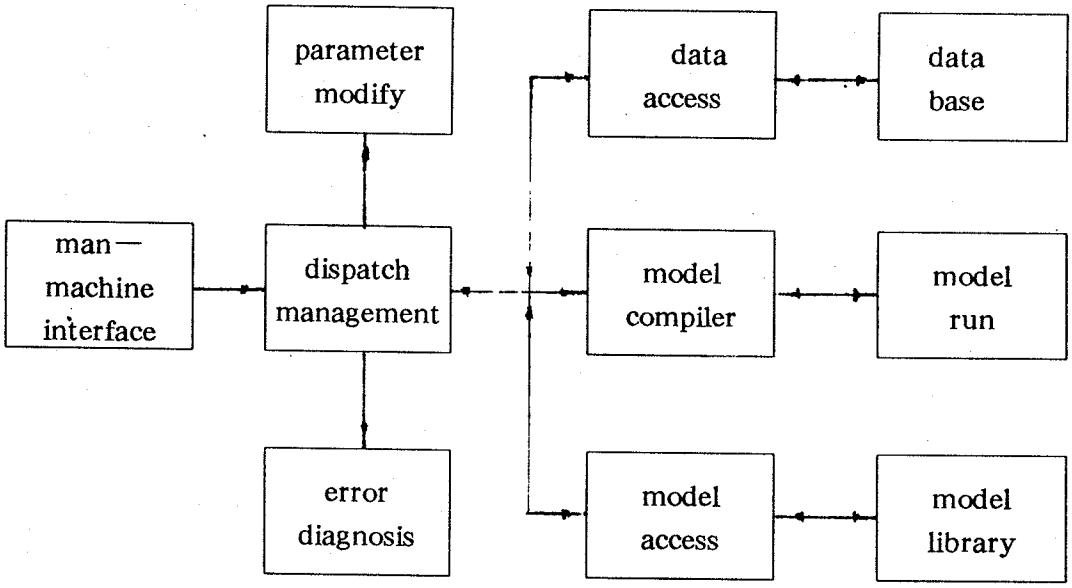


Fig. 1. DYNAMOX Overall structure

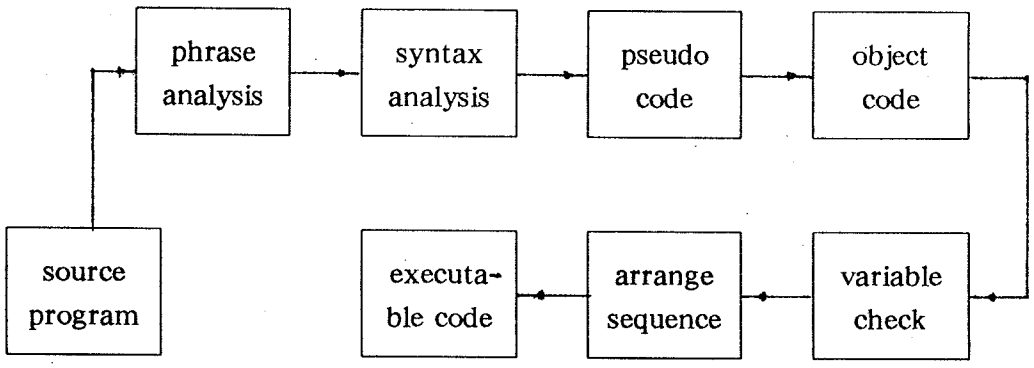
Dispatch Management and Man—Machine Interface

Man—Machine interface module recieves user's choice to DYNAMOX menu on full screen, and sends these commands to dispatch management module. According to the choice to system function, the dispatch management module harmonizes the operation of every module in overall structure. And controls system run . The module can not only combine the data from database wiht the model from model library into a run model , but the data and the model in the run model can be separated from each other , and sends them to database and model library respectively.

Model Compiler

The model compiler module is a nucleus in DYNAMOX, its performence affect directly the speed of compiling and runing for model. In order to enable DYNAMOX's compiler module program to be independent of a special computer, we use a medium langauge that is a executable code generated by compiling source program. Because DYNAMO language stipulates that a undifined variable can be first used, after medium code generate, it is necessary to check all being used ariables. Later on the medium codes are changed into object code again, and do arrangement sequence, last, executable code is got. This is the design thought

of the model compiler module program. The model compiler module program consists of phrase analysis 、syntax analysis 、 psedo— machine code generate、 object code generate、 variable check and automatic arrangement sequence module. The model compiler module is shown in Fig 2.



Figre . 2. compiler module flow chart

In fig. 2. , Source program means system dynamics simulation program written by user according to the syntax stipulated by DYNAMOX . For instance, (R RATE. KL =A. K * B. K +C. K) is a system dynamics simulation source program, sometimes, it is also called a system dynamics equation. Phrase analysis program reads in source program in one by one letter and one by one statement, and registers the type of every equation ,analyzes the equation phrase, checks whether the subscript of equation name acording with phrase regulation or not. When a equation name and its type are accoding with phrase regulation, dynamic memory allocation program will build a equation tree (i. e. the memory structure table)for the equation on the basis of the the equation name. Syntax analysis is an expression treatment program that analyses and treats the expression at right side in an equation. The syntax analysis program checks whether source program is according with syntax regulation or not. Because the expression at right side in an equation includes either arithmetic operation or logic operation, or both, the analysis algrithm for expression uses a operation sign priority rule of the expression — — oriented. During expression treatment, contant, variable and function can be treated by a semantics analysis program. First, the semantics analysis program takes out a operation sign and a operand from the expression in order, through analysis and treatment ,yields a operation instruction that is called medium (pseudo —Machine) code,at the same time,also builds a same medium code tree with the equation tree structure so that variable check and the generation of optimization objectve code can be done later on. Next, the operands that can not temporrarily become medium code are temporarily pushed in a operand stack, After the whole equation is treated, compiler stipatch program will treat next a equation continually. In this way,until meeting RUN equation or the source program end, syntax analysis of the source program for a model will be complet-

ed and all pseudo—machine codes are generated too.

For instance, in treatment equation (R RATE. KL =A. K * B. K+C. K) process, the semantics analysis program first generates two instructions stored variable A. K and B. K, and pushes the two address pointers stored the two temporary instructions into operation stack. later on,it generate a multiplication operation, the two address pointers kept variable A. K and B. K in operation stack are poped from the stack again, and, is changed into a medium code instruction for A. K * B. K. But the medium code instruction is only as temporary operation instruction in generation addition instruction, and its address pointer will be pushed into the stack, then generates another instruction stored variable C. K. Last, it generates a addition instruction, its two operands are the address pointer stored A. K * B. K in stack and the instruction stored variable C. K respectively. At this moment, the address pointer of the addition instruction is only pushed into the stack, and it is as the tree root of the medium code tree built by expression A. K * B. K +C. K .

After the medium code trees for all equations are correctly generated, variable verification starts. As mentioned above, because DYNAMOX's syntax stipulate that undefined variable can be first used, it is nesessary to verify whether all variables at right side of a equation appear at left side of the equation or not, and whether their subscripts accord with syntax regulation or not. In variable check process, variable check program first searches those medium code instructions stored variables. Next, by looking for the equation trees according to the variable name found already, seeks the equation of the same name to the variables. Then, obtains the permitable subscript of the variable by looking for variable table according to the equation and variable type found already, and compares the permitable subscript with actual subscript, if both subscripts are same, the variable has been defined, else will search continuosly . In this way, until all variables are checked and accord with stipulate.

The generation mode to objective code instruction is same with that of pseudo— machine code instruction. Before the interpreter program written in C language executes objective code instruction, the medium code instruction passed variable verification should first be transformed into objective code instruction. In transformation process,operands may be both immediate number、register and address pointer. Because of using the generation algorithm of optimization objective code tree and generating objective code chain by producing the objective code of medium code tree, the operation for a * b only generates a multiplication instruction , but it does not generate transmission data instruction, and the equation can be calculated as long as objective code chain is executed in order. In this way,it is both shorten the length of objective code chain and enhance the speed of treating equation largely. Arrangement sequence program provides automatic arrangement sequence function, user does not need to consider the programming sequence for source program. there are four kinds of equations that need.to arrange sequence. They are R equation、A equation、N equation and S equation.

Arrangement sequence is made by scanning objective code equation tree. After all equation tree are scanned once, if the equation to be arranged order can be found in the equations trees of unarrangement sequence, this arrangement order is successful, else , the arrangement sequence is failure. In arrangement order, the arrangement sequence program arranges same kind of equations at continuous place in arrangement sequence table, and every type of equation has its start address and value. After all equations are only arranged sequence successfuly, the simulation for the model can be done.

Model Run

The model run module is composed of three parts, i. e. run initiating program 、 algebra equation calculation and differential equation calculation program. Run initiaing program is used to set the initial value of zero for every equation except C (contant)、B(train)、SPEC and T(table) equation, and keeps the initial values of train variable and SPEC variable so that is used when rerun. Similar, it can also be used to set the initial value of unzero for every equation . Run initiaing includes both to initial the all system variables and functions, and it is mainly uesd to searche the address of the system variable and its value by scaning system variable table. The system variables includes simulation total time, integration step, integration mothod etc.

N equation、R equation and A equation calculation program mainly consists of the program of searching equation address、the program of researching equation chain code and the equation calculation program . First, the program of searching equation address seeks the address storaged the equation in stack by scanning equation tree, then, seeks the equation chain code according to the equation address found already, last, the value of the equation is calculated by equation calculation program . The calculation program for every kind of equation above mentioned is almost similar.

L equation calculation program: Because N, R, A equation is a kind of algebra equation, its calculation belong to algebra calculation. However, L equation is a kind of differential equation, its calculation is an integration calculation . In L equation calculation process, in order to calculate the vlue of L equation, it is necesary to dispatch relative integration rules, i. e. Euler、forth—order Runge—Kutta and Changeable step rules. Other calculation process is same with N, R, A equation calculation process.

Parameter Modification

During run , parameter modification module not only can modify any parameter in model and all system variables, such as constant, run time and output variable, but also change the output mode of each output variable, but not change integration rule. If the output mode for some variable has been changed , and the variables must be outputed in new output mode, it is necessary to rerun. However, because parameter modifying only changes objective code, rerun does not need to recompile source program. In this way, it enhances the

speed of the whole system run largely.

Model and Data Access

Model access module is a model library management program. It has two functions. First function is to storage user's simulation model to model library. Another function is to take out the relative simulation model from the model library and send it to the dispatch management module. Data access module is a database management program, it can storage the data produced by running simulation model into database . In order to output simulation result, it is necessary to take out the data from database. sometimes, when the initial condition of a model equation needs the simulation result of another model, in the data format stipulated by DYNAMOX and ALRACL database management mode , the data access module accesses the database.

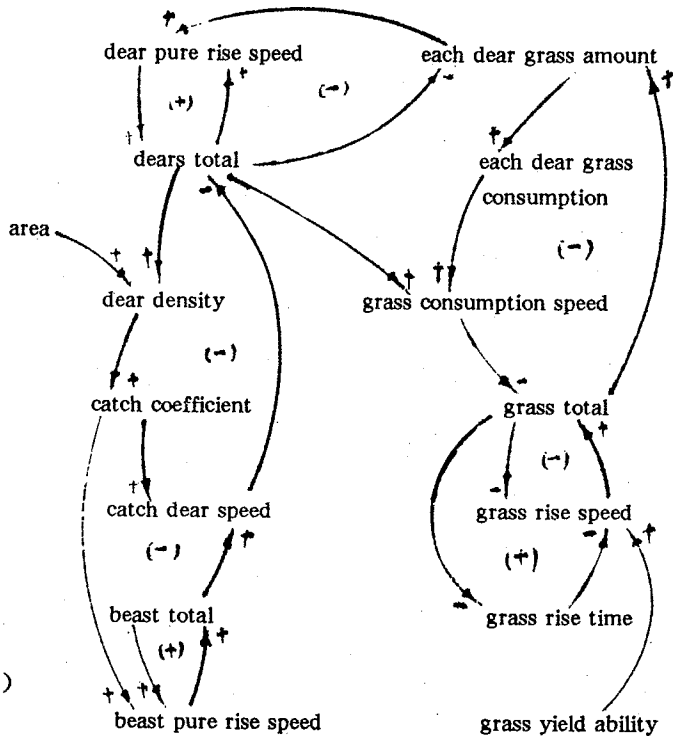
EXAMPLE

We simulated many system dynamics models, such as world model (world dynamics i. e. W5 and W2) and the planning model for some area (over 3200 equations) and so on . In addition, in order to illustrate our DYNAMOX model size, we simulated a pure mathematics model with 10000 difference equations that have not practice sense. Here , we give a example about the organisms relationship model between deers, beasts and grass in Kaibab plateau . Its system dynamics flow chart is, and the simulation result in figre. 4 , and include DYNAMOX source program .

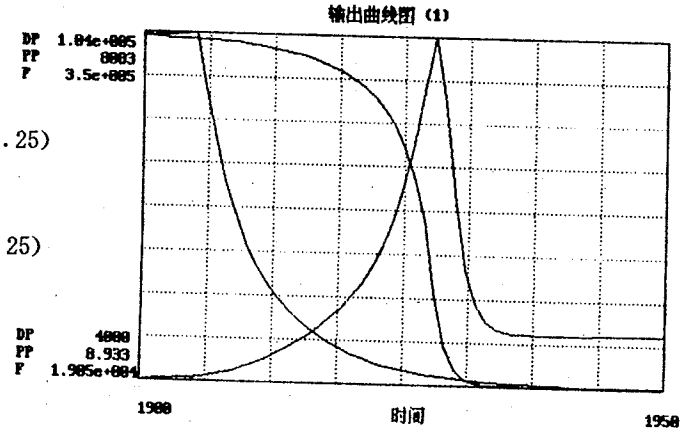
CONCLUSION

DYNAMOX is a system dynamics simulation software suitable for microcomputer on XENIX enviroment, and it has been successfully used in practical system. particlarly, has been used in "the decision support system about harmonizing science、technology、economic and society development in SHAAN XI province". Many users think that DYNAMOX's sperformence is superior to earlire system dynamics simulation language in many aspects. Its popularization will certainly promote the study and application of system dynamics simulation technique in many fields, Paticularly, in the decision support system about harmonizing science、technology、economic and society development planning.

```
NOTE DEER
L   dp.k=dp.j+(dt)(dngr.jk-dpr.jk)
N   dp=dpi
C   dpi=4000
R   dngr.kl=dp.k*dgrf.k
A   dgrf.k=tabhl(tdgrf,fr.k,0.25,2,.25)
T   tdgrf=-.75/-.5/-.25/0/.12/.2/.23/.24
A   fr.k=fpd.k*nfpd
C   nfpd=1
A   fpd.k=f.k/dp.k
R   dpr.kl=dkr.k*pp.k
A   aa.k=dp.k/area
C   area=800000
A   dkr.k=tabhl(tdkr,dd.k,0,.1,.01)
T   tdkr=0/.2/1.2/3.2/5.4/7.6/8.6/9.3
X   /9.8/10/10
NOTE BEAST
L   PP.K=PP.J+(DT)(PNGR.JK-PBR.JK)
N   PP=PPI
C   PPI=8000
R   PNGR.KL=PP.K*PGRF.K
A   PGRF.K=TABHL(TPGRF,DKR.K,0,.6,.1)
T   TPGRF=-.04/0/.02/.035/.045/.05/.05
R   PBR.KL=PP.K*STEP(RF,RST)
C   RF=.2
C   RST=1905
NOTE FOOD
L   F.K=F.J+(DT)(GR.JK-FC.JK)
N   F=FI
C   FI=350000
R   GR.KL=(FCAP-F.K)/FRT.K
C   FCAP=350000
A   FRT.K=TABHL(TFRT,F.K/FCAP,0,1,.25)
T   TFRT=20/8/3/2/1
R   FC.KL=DP.K*FCPD.K
A   FCPD.K=TABHL(TFCPD,FR.K,0,1.5,.25)
T   TFCPD=0/.25/.5/.75/1/1.12/1.2
NOTE CONTROL
SPEC  DT=.1
SPEC  LENGTH=1950
SPEC  PRTPER=1
SPEC  TIME=1900
GRAP  DP/PP/F
RUN
```



Figre . 3. System Dynamics flow chart



Figre . 4. Simulation Result

REFFERENCE

Wang, H. B. 1989. System dynamics guide ,Energy publishing house.

Aho,A. V. 1976. Optimal Code Generation for Expression Trees. Journal of the Association for Computing Machinery vol. 23. NO. 3 pp 488 —507.

Addisson Weslay Publishing Company Inc. 1983. User Guide and Refference Manual for Micro——DYNAMO System Dynamics Modelling language IBM—PC Version,

Uhran J. J. 1984. The Structure of NDTRAN—— a system simulation language, IEEE TRANSACTION ON SYSTEM VOL SMC—14 NO.

Phillips. J. B. 1978. Threaded Code for Laboratory Computers, SOFTWARE —PRACTICE AND EXPERIENCE,VOL. 8, PP257—263.