WORKING PAPER

RECENT DEVELOPMENTS IN
SYSTEM DYNAMICS SOFTWARE

István Vályi

November 1987
WP-87-110

IIASA
International Institute
for Applied Systems Analysis

NOT FOR QUOTATION
WITHOUT THE PERMISSION
OF THE AUTHOR

RECENT DEVELOPMENTS IN SYSTEM DYNAMICS SOFTWARE

István Vályi

November 1987
WP-87-110

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
2361 Laxenburg, Austria

FOREWORD

This paper is a short review of a conference held in Sevilla, Spain, in October 1987. Organized by the *Systems Dynamic Society*, it concentrated around concepts in methodology and applications of nonlinear system modelling within the framework introduced by Jay Forrester and his followers.

The attitude to this approach is controversial. For example, the respective methodologies do not involve the identification of system parameters and the construction of the models from available data does not involve and may even contradict with the rigorous concepts and techniques of modern system theory.

The review given here does not discuss the relevance of "System Dynamics". It merely gives some information on the topics presented at the conference in Sevilla.

A. Kurzhanski
Chairman
System and Decision Sciences Program

RECENT DEVELOPMENTS IN SYSTEM DYNAMICS SOFTWARE

István Vályi

1. INTRODUCTION

This paper is to report recent results in system dynamics software development relying mainly on the material presented at the 1986 International Conference of the System Dynamics Society held under the title 'System Dynamics on the Move' [1]. The conference took place between 22-24 October 1986 in Sevilla, Spain. Participants came from more than twenty countries from four continents and included leading personalities of the system dynamics community.

System dynamics was developed at the end of the fifties by Jay W. Forrester to assist in the study of complex systems a rather wide range of users, with little or no mathematical background. According to D. H. Meadows ([6] p. 31.): dealing with questions about the dynamic tendencies of complex systems, that is, the behavioral patterns they generate over time. System dynamicists are generally unconcerned with precise numerical values of system variables in specific years. They are much more interested in general dynamic tendencies; whether the system as a whole is stable or unstable, oscillating, growing, declining, or in equilibrium.' In mathematical terms this means that in system dynamics we use systems of ordinary nonlinear (functional) differential equations for representing reality, and questions and answers are typically formulated in qualitative terms. As is well known, the mathematical theory that deals with such problems is rather difficult.

G. P. Richardson and A. L. Pugh III, [7], identify seven steps in approaching a problem from a system dynamics perspective:

problem identification and definition

system conceptualization

model formulation

analysis of model behavior

model evaluation

policy analysis

model use or implementation.

This procedure usually involves passing through three stages of model representation: causal-loop diagrams, flow diagrams and equations. The first stage in developing the formal model is to draw causal-loop diagrams. Causal-loop diagrams are directed graphs where nodes represent elements of the system (i. e. variables of the model) and edges the causal relations connecting them. Edges are marked with positive or negative signs to indicate whether increase causes increase or decrease. The second, intermediate step is taken by drawing the flow diagram, that is to classify variables in the causal loop diagram according to their roles in the model (like levels, rates, constants contain all the information needed to write the equations, that is the final, analytical form.

Despite frequent criticisms the technique has been widely applied, and nowadays is considered as a profession. It has proved to be specially useful in education and combined with gaming techniques. Arguments were focussed around the following points:

problems related to qualitative validation, sensitivity and parameter tuning ('lack of empirical base', Oerlemans et al., [5])

unwarranted use of mathematical objects that are sometimes very difficult to handle (for alternatives see e. g. J. Talavage, [4])

In standard system dynamics, the solutions of the system of differential equations are computed by way of approximating the integrals by Euler's method with a given interval length, i. e. considering the right hand side constant in each interval. This procedure is realized in DYNAMO, [7], a computer simulation language developed at MIT, that has been associated with the method from the very beginning. Even the analytic form of the models, i. e. the differential equations are normally formulated in this language, rather than in mathematical notations. Since the mid sixties DYNAMO is based on FORTRAN. Its latest versions like DYNAMO III and IV became rather flexible as they are capable to handle subscripted variables and DO loops, accept user defined functions, or offer a possibility of selecting between various integration methods. Mini-DYNAMO is a simplified version that runs on small computers with 20K core memory while Micro-DYNAMO and Professional DYNAMO are intended for professional personal computers like the IBM PC. DYSMAP, developed in the seventies at Bradford University, UK, [3], and NDTRAN at Notre Dame University, USA are widely used as well. These are extensions of DYNAMO and offer better graphical output, and more sophisticated options for model checking and analysis. A significant development is STELLA, [8], a program running on the Macintosh, fully exploiting its excellent graphics facilities. STELLA starts with defining the model in the form of flow diagram on the screen, and develops the analytical form automatically. Another interesting system is DYSYS, [2], developed at the University of Kassel, FRG, that is a simple simulation language written in BASIC with minimal hardware requirements, very well suited for educational purposes.

The results presented at the conference reflect the recognition of the problems brought up by the critics, as well as the efforts to ease further the modelling procedure.

This report is divided into two parts, one dealing with traditional type software, and the other with expert systems techniques.

2. NEW SYSTEM DYNAMICS SOFTWARE

We start this section by a new implementation of a the simulation language DYSMAP. The new version called DYSMAP2, developed by University of Salford Computer Services Section, UK, is an extension of DYSMAP, i. e. it accepts any DYSMAP program with some minor modifications. DYSMAP2 is intended for use on microcomputers and is based on interpretation at runtime rather than the usual technique of translation into FORTRAN. This results in:

the possibility of handling runtime (in addition to syntax) errors,

in cases more than ten times faster execution and improved interactive possibilities.

An important possibility for checking the correctness of the model is through the dimensional analyzer. The interactive setup means that an interactive session starts with a basic run of a model and then the user is allowed to select any form of output (including colour plots of any variables, numerical values on the screen, or in files in ASCII format), then parameter values and even equations can be modified and the model rerun. Data communication through files is possible in two directions. Therefore the interactive environment makes it easy to apply optimization for parameter tuning or sensitivity analysis. It is in fact planned to include this feature into the package. In addition to computed results, other supplementary information is available through the HELP option.

A related paper by R. Keloharju (Helsinki School fo Economics, Finland) and E. F. Wolstenholme (University of Bradford Management Centre, UK) presented a new version of DYSMOD (Dynamic Simulation Model Optimiser and Developer) that was developed in the late seventies as a supplement to DYSMAP. The software is currently under further development by Bradford and Salford Universities, UK, in order to be combined with DYSMAP2. DYSMOD uses a hill climbing routine for optimization. The DYSMAP simulation run is used in the context of the optimization routine to compute the value of the goal function. According to practical experience, a rather large number of iterations (100 or more) is needed to achieve the optimal values, but the individual iterations are not very expensive.

The authors list the following uses of optimization in system dynamics models:

fitting the model to past data,

finding desirable parameter (and table function) values for better performance of the system,

finding desirable decisions, structure, or simplification, by taking e. g. convex combinations of expressions, that represent alternative policies, at the right hand side of equations, or forcing the effects of certain parts of the model to zero,

sensitivity analysis, by relaxing the found optimal value and checking the largest possible change in key parameters.

Unlike the traditional ways of carrying out these tasks, the method allows all this to be done simultaneously, a feature that constitutes a major conceptual advantage. The structure of the program is shown in the Figure 1.

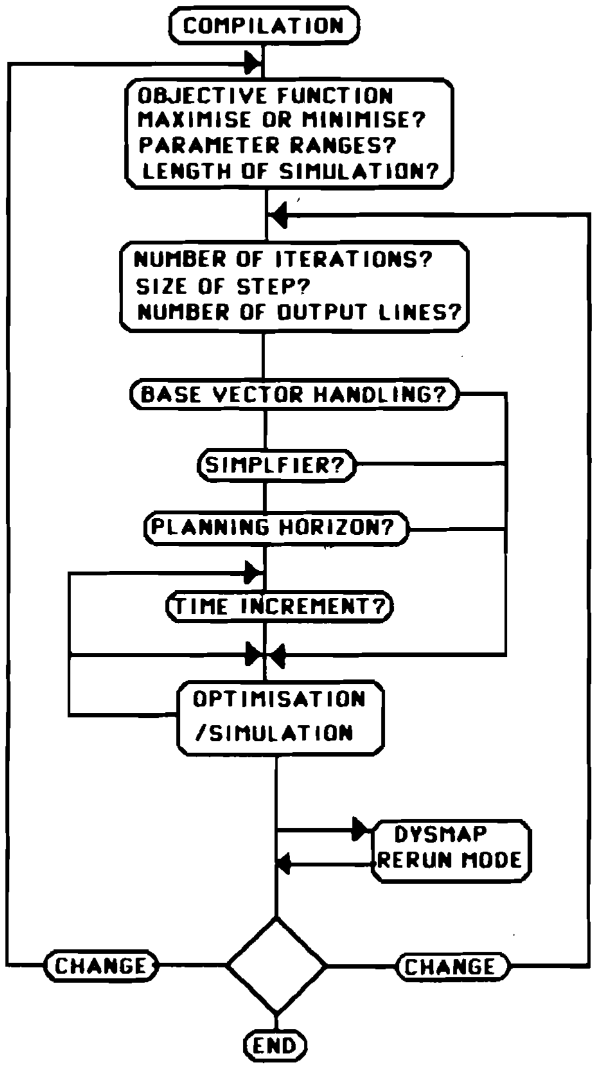
COMPILATION
OBJECTIVE FUNCTION
MAXIMISE OR MINIMISE?
PARAMETER RANGES?
LENGTH OF SIMULATION?
NUMBER OF ITERATIONS?
SIZE OF STEP?
NUMBER OF OUTPUT LINES?
BASE VECTOR HANDLING?
SIMPLFIER?
PLANNING HORIZON?
TIME INCREMENT?
OPTIMISATION
/SIMULATION
DYSMAP
RERUN MODE
CHANGE
CHANGE
END

Figure 1.

The presentation by H. Krallmann, B. Rieger and G. Hentzelt of the Technische Universitaet (West-)Berlin reports about a system that constitutes a user friendly communication and control system for existing system dynamics models implemented on a mainframe computer. It should be pointed out here that the system is universal in the sense that it could be easily extended to handle other types of models as well. Its major aim is given by the authors to assist in:

selecting preformulated models

generating alternatives by changing model parameters

controlling simulations (storage and handling of model results)

graphic processing of results

providing model documentations.

Currently the system consists of a library of DYNAMO models, a (commercial) statistical analysis package called SAS 83 to perform the graphical representation of results and their own control program. The operation of the package is done through menus and is supported by the HELP feature. Its structure shown in Figure 2.

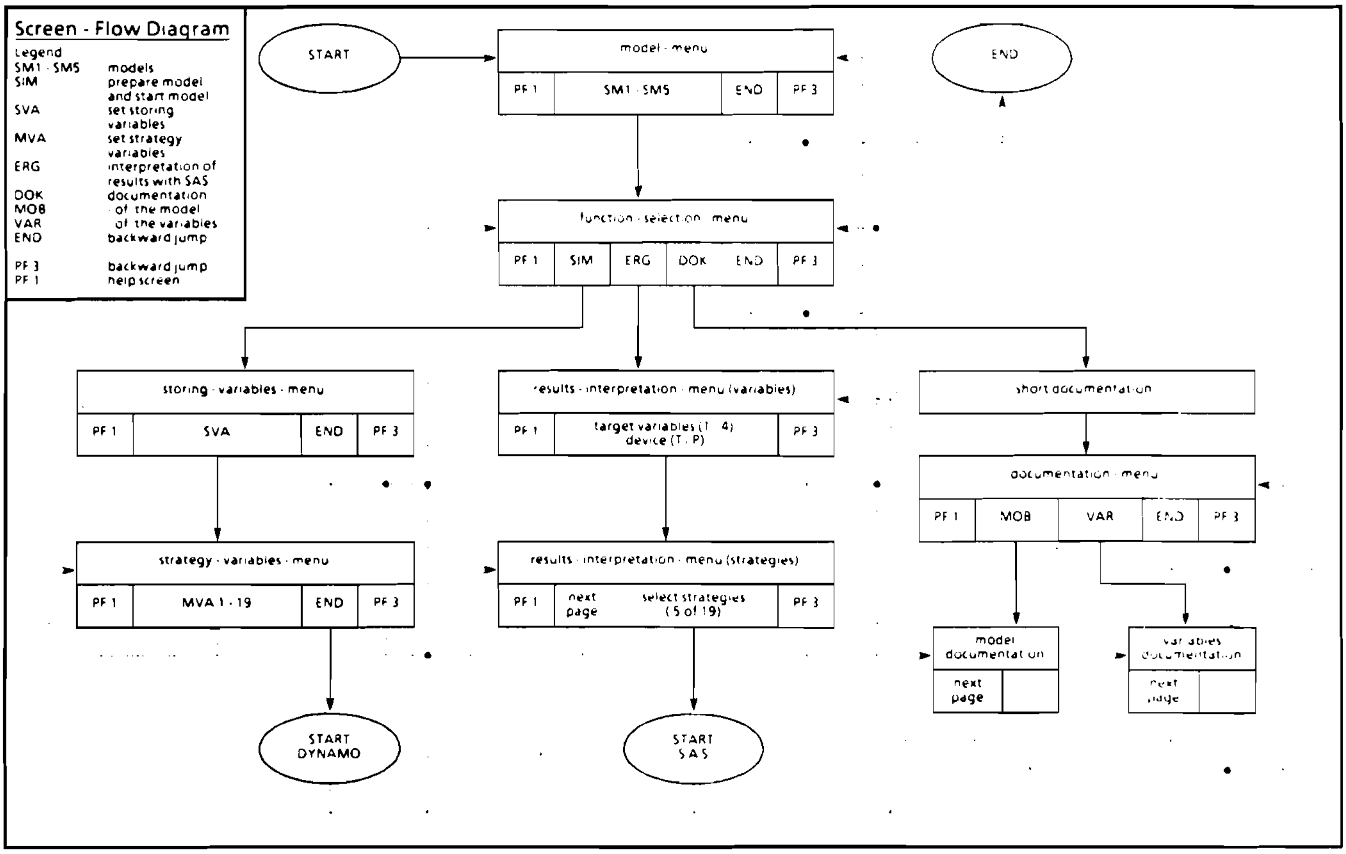

Figure 2.

Finally we mention the piece of software called SDSE (System Dynamics Sofware in Education) by A. Toval, A. Requena, S. Martinez and J. Monreal of the Universidad de Murcia, Spain. As the name indicates the system was designed to facilitate the teaching and use of System Dynamics in education, namely in schools and high schools. The system was written in BASIC and was implemented for IBM PC (compatibles) under MS DOS.

3. EXPERT SYSTEMS FOR SYSTEM DYNAMICS

The expert system type software to be introduced here have the general aim to cover, in addition to the numerical and administrative type operations, as much of the system analysis process as possible.

This means, in more detail, in the system dynamics context to include:

the specification of variables and implications connecting them to establish the system boundaries and the causal diagram,

classification of the variables to construct the flow diagram,

writing the model equations,

and (of course) simulation, output and documentation.

Let us start with EASDM, or an Expert Aid for System Dynamics Modelling that was presented by J.C. Gonzalez and G. Fernandez from the Laboratory f Cybernetics and Systems Theory of the Polytechnical University of Madrid. The major novelty of the system is that it is able to carry out the third step of the previous list, that is the construction of the flow diagram in a semiautomatic way. EASDM was programmed in PASCAL and PROLOG for personal computers with MS DOS operating system. (PROLOG can nowadays be regarded as the standard language for 'knowledge representation' that is for logical programming.)

EASDM has a modular structure consisting of the following units:

definition module, to introduce variables and logical connections,

classification module, to convert the above into a flow diagram,

equation module, to write the formal model,

processing module, to translate the formal model into a PASCAL program,

simulation module, to perform computations, and convert the results into the desired numerical or graphic form.

The functioning of the classification module is based on the observation, and the related theory, that a correctly drawn causal loop diagram already contains all the necessary structural information for drawing the flow diagram. Correctness is an essential feature here, that is ensured by the diagnostic capabilites included into the module.

Another system with good supporting capabilities in the design of the model is LSC, a Continuous Simulation Software developed in the Instrumentation and Control Laboratory, Algeria presented by M. Gaci and A. Babaamer. The system is partly implemented on a mainframe computer and is written in PASCAL and FORTRAN.

The software leads the user through the stages of modeling as indicated in Figure 3.

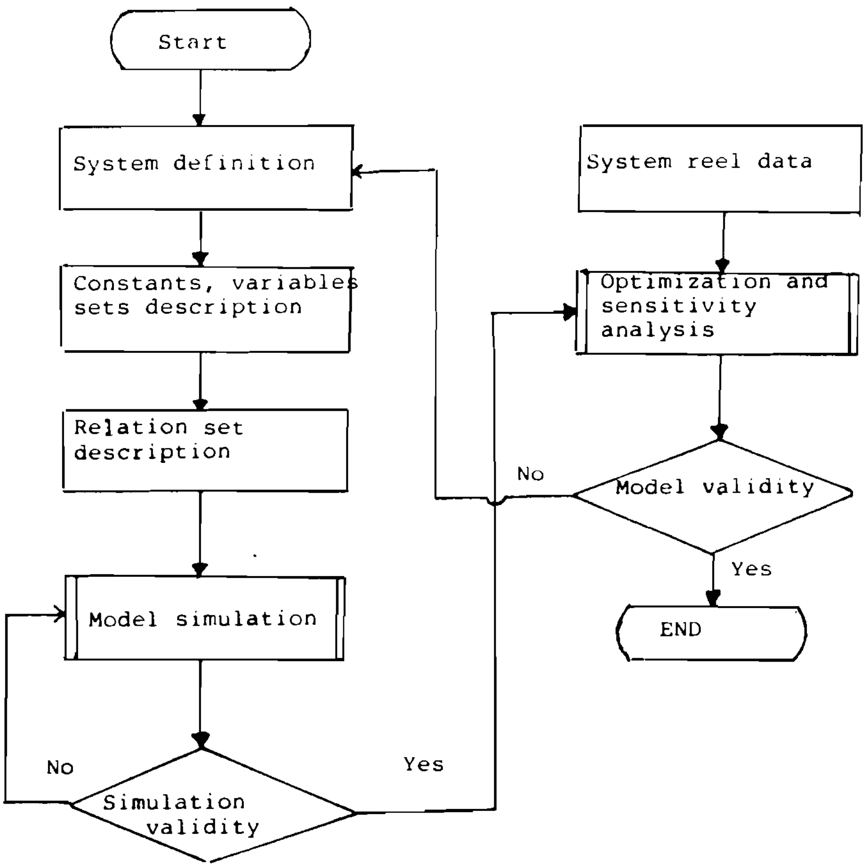

Figure 3.

The system works interactively, using a high level language that with the following set of 'environments':

starting environment: model name and simulation type

formal declaration environment: regrouping commands, defining variable types

initialization environment: defining initial values, constants etc.

relations environment: defining equations

edition environment: editing simulation results and output

control environment: checking the coherency of the model

Model definition is carried out by passing perhaps several times through these environments.

The software uses integration by Gear's algorithm for stiff systems. For nonstiff systems the modified Hamming and Runge Kutta Gill algorithms or simply Euler's method can be used. A special option is available for discrete systems. It is planned that a sensitivity analysis module be included, computing a sensitivity coefficient according to p. M. Frank.

Further modules of the system are (not fully implemented yet):

compilation module: lexical and syntactic analysis, and code generation

formal derivation module

data structure module

library module

graphics module.

Finally we consider a contribution that we regard as the most substantial among those reported in this review. The system is called MAPS (Mathematical Advising Production System), and is an expert advisor for the qualitative analysis of dynamical systems. In its present form MAPS deals with second order autonomous systems of ordinary differential equations. It is implemented in a prototype version on an IBM PC. Minimal RAM required is 512 Kb. The system is written in muSIMP, i. e. the host language of muMATH and uses the latter for carrying out symbolic calculations.

Based on the qualitative theory of differential equations, MAPS studies the asymptotic behaviour of dynamical systems, performing a parametric analysis of the nature and stability of the asymptotic states. It is pointed out that under certain conditions, larger complex systems can be reduced to simpler (deterministic) ones in such a way that from the qualitative behaviour of the latter conclusions can be drawn for that of the former. In this way the software provides us, although in a very limited range, with a mathematically correct answer to some of the questions usually posed in system dynamics. The analysis cannot be made completely automatic, and it needs an interaction between the system and the modeler. According to the authors a final system should incorporate, along with formal mathematical knowledge, some heuristics, as well.

The currently implemented version studies systems of the form:

$$\dot{x} = f(x, y, p, \bar{c})$$

$$\dot{y} = g(x, y, p, \bar{c})$$

where x and y are scalar state variables, $\bar{c}$ is a constant vector and p is the scalar parameter along which the analysis is performed.

The steps of analysis:

verification whether the system belongs to some special class (like linear, hamiltonian etc.)

search for equilibrium points

linear stability analysis for each equilibrium point as a function of the parameter p.

In order to complement analytical results, numerical simulations can also be performed.

4. REFERENCES

1. ARACIL, J., J. A. D. MACHUCA, M. KARSKY (Eds) *System Dynamics: on the Move* Proceedings of the 1986 International Conference of the System Dynamics Society, Sevilla, Spain, (1986)

2. BOSSEL, H., *Umweltdynamik - 30 Programme fuer kybernetische Umwelterfahrungen auf jedem BASIC-Rechner*, TeWi Verlag, Munich 1985.

3. CAVANA, R. Y., R. G. COYLE, *DYSMAP User Manual*, University of Bradford, (1982)

4. LEGASTO JR, A. A., J. W. FORRESTER, J. M. LYNEIS (Eds) *System Dynamics*, North Holland, Amsterdam, New York, Oxford (1980)

5. OERLEMANS, T., M. TJELLINGS, H. DE VRIES *World Dynamics: Social Feedback May Give Hope for the Future*, Nature 238, pp 251-255. (1974)

6. RANDERS, J. (Ed) *Elements of the System Dynamics Method*, The MIT Press, Cambridge, Massachusetts, USA, (1980)

7. RICHARDSON, G. P., A. L. PUGH III, *Introduction to System Dynamics Modeling with DYNAMO*, The MIT Press, Cambridge, Massachusetts, USA, (1981)

8. RICHMOND, B., *A User's Guide to STELLA*, High Performance Systems Inc., 13. Dartmouth College Highway, Lyme, New Hampshire, USA, (1985)