SDSE, SYSTEM DYNAMICS SOFTWARE IN EDUCATION

Toval, A.(1); Requena, A.(1); Martínez, S.(3); Monreal, J.(2)

(1) Escuela Universitaria de Informática. Universidad de Murcia
(2) Facultad de C. Económicas. Universidad de Murcia
(3) Instituto de Economía Agraria y Desarrollo Rural. Consejo de Investigaciones Científicas (CSIC). Madrid

Abstract. SDSE is a microcomputer based, interactive and inte-- grated system designed to facilitate the teaching and use of Sys tem Dynamics in Education. In this way teachers and students / without extensive Computer training are able to construct and si mulate system models after short period of learning.

System Dynamics has been included in Atenea, a Spanish Ministry/ of Education and Science project in order to introduce computers in Education. Nowadays, SDSE is being used as a supporting pro-- gram to teach System Dynamics to school and high school teachers in Atenea project courses run by Comunidad Autónoma in Murcia.

In this paper, functional characteristics of the system are des- cribed, at the same time that SDSE is applied to build and simu- late the simple and classical example of the inventory control / system. Nowadays, SDSE is programmed in FBASIC (FUJITSU FM/7) and GWBASIC (IBM PC and compatible)

INTRODUCTION

The availability and new flexibility of modern microcomputers / has given a boost to system dynamics. Powerful tools and langua- ges have been developed to simulate and analyze more or less ge- neral systems in many different areas, thus favouring the gro- wing use of the subject.

Education is not an exception to this fact, even more so nowa- / days when the use of computer simulation in instruction is part/ of the recommendations for computer applications in this area in many approaches to the use of computers in Education. Zinn(1973) Breuer(1985), Eyferth(1974), Atenea(1985), Balkovich(1985).

Spain, like other countries interested in updating the old educa tional system is carrying out a project to introduce the new in- formation technologies at elementary and secondary school level. Atenea project(1985) as it is called, has been designed by a / work-group of the spanish Ministry of Education and Science.

Atenea project is concerned with simulation and, in particular , system dynamics in several aspects that we summarize as follows

1.- The specific simulating languages and programs development/ is essential to permit the teachers to create their own models or , at least, to adapt or to modify those already im-- plemented by others.

2.- Use of system dynamics on the part of teachers and students/ is thought as useful for building models. Especially social and economic ones. To achieve these aims it will be desirable to have available a single tool which facilitates the / use of these techniques.

3.- Educational games use is recommended both to permit students to use different strategies or policies to confront problems and to introduce them to decision making.

SDSE has been designed and implemented bearing in mind these requirements. Further, it tries to close the gap caused by the / lack of specific software in the spanish language, adequate for education system dynamics. This is a problem not only in Spain / but also in another spanish-speaking countries.

Another specific languages such as DYNAMO, MICRODYNAMO, DYSMAP or DYNAMINE, are well-known but we think that their use in education has some problems mainly derived from the efforts of lear-- ning a programming skill using a not very natural language . Moreover, initially, these languages were professional-use oriented, which makes difficult their adaptation to a pedagogical use

Although SDSE is based on some DYNAMO concepts too, it has been designed and developed exclusively for an educational use. In / this paper, functional characteristics of the system are described, including:

- Model equations editing and management in user language
- Scenarios editing and management
- Simulation
- Results management
- Aids library, error messages, etc.

Along with the description, we present an application of SDSE/ to a typical example: the inventory control system, in the sim-- plest case, to be able to show most of the system features aro-- und the process of models building and operating, and scenarios or "images" management (we consider an "image" as the set of all the values of the variables along the simulation time horizon in relation both to a given model and linked scenario).

Finally, we´ll speak about limitations of the actual system and future improvements and extensions of the package.

SDSE DESCRIPTION

SDSE is a microcomputer based, interactive and integrated system designed to facilitate the teaching and use of system dynamics / in Education. It is implemented for FBASIC (FUJITSU FM/7) and GWBASIC (IBM PC and compatible) under MS DOS. The system is menu-driven, which makes it easy to use and keeps the user aware / all the time of wchich section of the program where he is in.

SDSE is subdivided into modules, each of which operates as an independent functional component of the overall system. So it is possible to focus attention on each component as a separate smaller problem

Another advantage of modularity is valuable during system maintenance. When a system is implemented as a single unit, it is of-

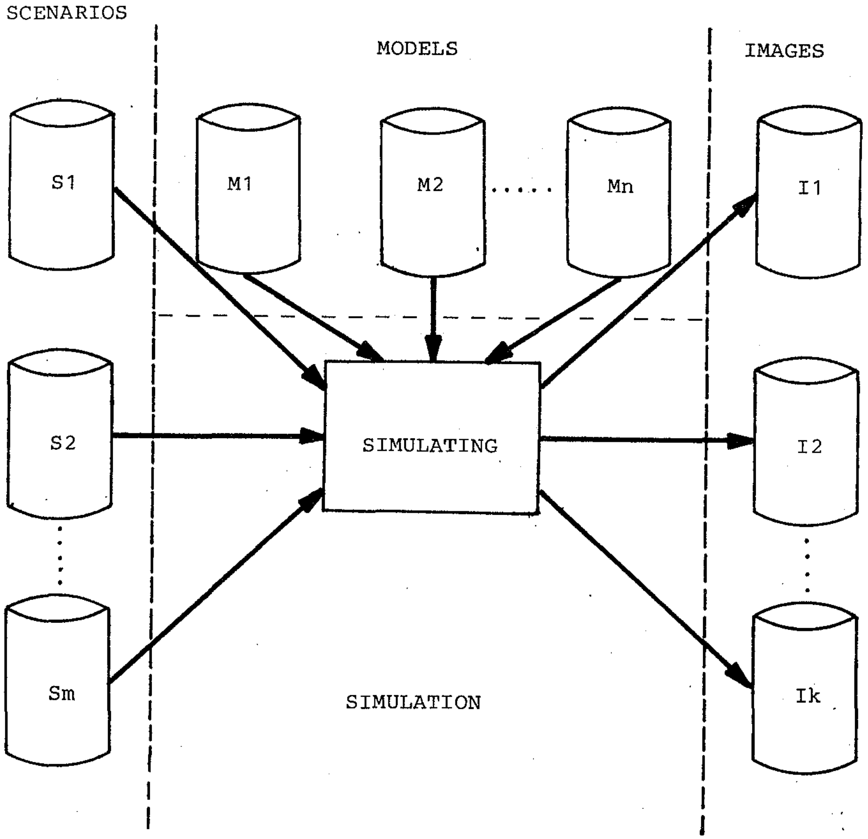


Fig. 1 SDSE General Diagram

ten difficult to change it because interaction between statements, variables and scenarios complicates changes and makes / it easy to introduce errors. When the system has been divided/ into modules, changes and corrections are much easier to im-- plement . Golden(1985)

As shown in Fig. 1, the main modules are: Models, Scenarios , Simulation and Images.

MODELS

In spite of the profusion of text editors existing on the market, we have prefered to build a specific one to edit systems/ dynamics equations in user language, which later will be automatically converted to a set of instructions which will be/ runnable by means of the simulating module. The editor provides some facilities and a direct connection to the other modules.

SDSE equations syntax is similar to DYNAMO, but simpler in some aspects: neither timescripts nor variable type specifica- / tion are necessary at this point. Relationships between variables are functionally expressed, in the manner they are obtained from the DYNAMO diagram by the modeler.
Two kind of structure are used to write the models: sequential and alternative. Repetitive structure is not available in this release, but it will be subsequently implemented.

The syntax for the equations may be:

1.- <Variable >= <Expression >

2.- SI <Variable or Expression>< Logical connective >
   < Expression >
   HAZ < Variable> =< Constant or Variable or Expression >

(spanish SI..HAZ.. is the same as IF.. THEN..)

Where <Variable >is a level, rate exogenous or auxiliary.

<Expression > is any mathematics expression composed of < Variable >, constants, functions, tables or delayed variable of the associated scenario. <Logical connective > may be one / of the next:

- "< " : less than

- "> " : bigger than

- "< =": less or equal to

- "> =": bigger or equal to

- " = ": equal to

- "<> ": different to

Figures 2 and 3 show respectively, the inventory control loop and flow diagram.

Associated SDSE equations should be:

```
INV=INV+DT*(OR-SR);
SI T>=4 HAZ SR=20;
DISCR=DINV-INV;
OR=FOW*DISCR;
@
```

where DT is the time interval to compute the equations.

To write variables wchich are computed by means of a table / (in relation to the time or to another variable) we should / use the notation:

< Variable > $

The corresponding table of values is previously defined, only once, using the scenarios module, because conceptually the table as a set of values, belongs to the scenario of simulation

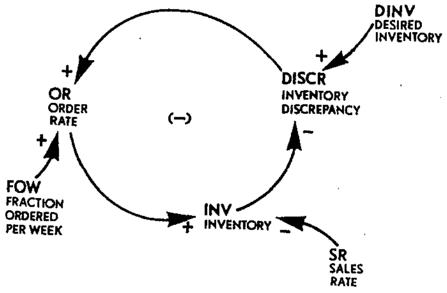


Fig.2 Inventory control loop

So, each time that the / user needs to write the variable in the model, he will do it using this single notation.

To keep previous timestps/ values of a variable (a delayed variable), the following notation is used:

<Variable> # N

where N indicates the number of previous timesteps/ to the present time. So , V#2 indicates the value of

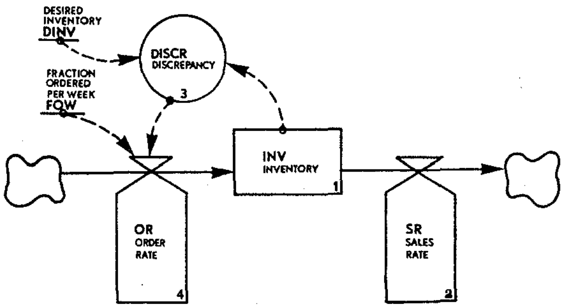


Fig. 3 Inventory flow diagram

V two iterations ago. In this way we can actively store previo us values of the variable to be used in any place of the mo-- del,where necessary.

As an example, we can suppose that SR flow takes values by means of a table and OR is two timesteps delayed, then:

```
INV = INV + DT*( 0.5*OR#2  + 0.5*OR  - SR$(T) )
```

means that the input flow to the store, in time Tk, is one / half of OR value, two iterations ago, added to one half or OR actual value; output flow is the SR table value corresponding/ to the actual entry value, Tk (or interpolated value when ne- cessary).

On the other hand, the mathematical functions available in / SDSE are the same as one can use in FBASIC, for FUJITSU FM/7 , and GWBASIC for IBM PC, or compatible, because the compiler / keeps them unchanged when creating the object model.

Model editor operation is similar to the other text editos in the market. However,some facilities are available, it is possi ble to:

- Show models and linked scenarios
- Show linked tables
- Display visual aids
- Chain to the other modules of the package
- Compile a model in relation to a linked scenario, etc.

We shall consider compiler separately.

Compiler

Although we are using the word "compiler", this part of the model module is really a BASIC sub-program generator.

To compile the user language model, source model, to runnable/ model, object model, the computer asks both for the source model and the chosen linked scenario names. Compiler uses a push-down automaton to recognize and check for equations or / scenario errors. It verifies that each variable of the model has previously been created by the linked scenario and checks/ that expression of the variables in the model are in accordan- ce with their type, indicated by the scenario.

Then two possibilities can ocurr: fistly, when some errors in the source model have been found, an error listing is obtained , where is accurately pointed out the place and type of errors Secondly, if errors have not been found, compiler generates / the object model as a BASIC subprogram, converting user varia- ble names to array elements and model equations to BASIC ins- tructions, which will be, later, runnable by simulating module by means of merging the object model to the general simulating program (Fig. 4).

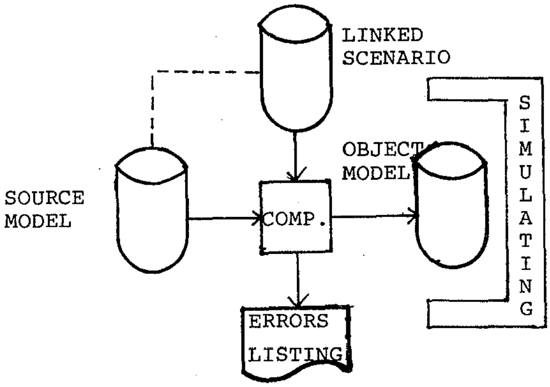


Fig. 4 SDSE Compiler

SCENARIOS

We consider the scenarios composed by the next elements:

- Time horizon: initial time, final time, time interval (DT), printing and plotting interval (for numerical and graphic output).

- Variables specification: names and type of levels and its/ initial values, rates, auxiliaries, constants and its values.

- Tables definition

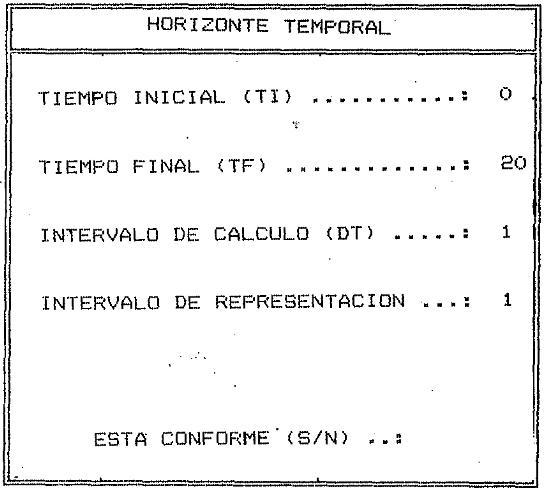
- Delays indication and initial delay values

Tables use linear interpolation in a similar way as DYNAMO / does.

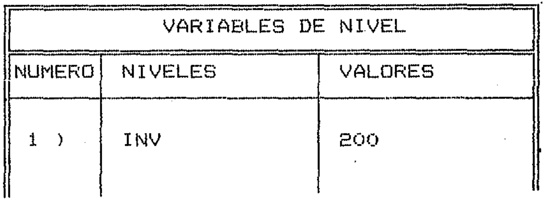
Once scenarios are created, they are disk stored as files. / Their management is the typical one for a file management system, you can:

- Build and store new scenarios
- Display and maintain stored scenarios
- Delete scenarios and tables
- Link scenarios to models or tables to scenarios

The SDSE scenario for the inventory control system should be:

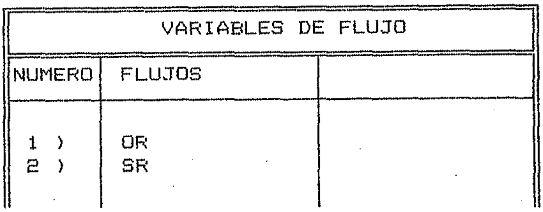


```
                 HORIZONTE TEMPORAL

TIEMPO INICIAL (TI) ...........:   0

TIEMPO FINAL (TF) .............:   20

INTERVALO DE CALCULO (DT) .....:   1

INTERVALO DE REPRESENTACION ...:   1


      ESTA CONFORME (S/N) ..:
```



VARIABLES DE NIVEL

NUMERO	NIVELES	VALORES
1)	INV	200



VARIABLES DE FLUJO

NUMERO	FLUJOS	
1)	OR	
2)	SR	

'Auxiliaries and constants for the proposed scenario would be:

VARIABLES AUXILIARES		
NUMERO	AUXILIARES	
1)	DISCR	

TASAS		
NUMERO	TASAS	VALORES
1)	FOW	0.5
2)	DINV	200

SIMULATION

A compiled model may be run in relation to a scenario, chosen among the possible one linked to it. Optional features in the module are, to:

- Tabulate numerical output, on the screen, of the results / for the selected variables (see table 1).

- Tabulate numerical output, on the printer, for all the va-- riables

-Change the active scenario or the active model.

- Display/Change active scenario parameters.

At this point, changes in parameters are temporary, they are/ in effect only during the run in which they were made. Permanent changes, recorded in scenario files, are not possible in this module, they are allowed exclusively by the scenario module.

Each run automatically generates an image as a file on the / disk. Later, it could be graphically or numerically managed using the adequate module.

Four images , linked to the same model, may be simultaneously active. As fig. 5 shows below, images may be originated by the model run on different scenarios (S1,S2) or by the model run only in respect to an original scenario over which some interactive changes in its parameters have been made (S1',S2')

To simulate any compiled model the process consist of merging the "object" model (a BASIC subprogram) with the general simulating module and loading of the chosen scenario. If some error ocurrs during the run it will be identified as a BASIC running time one, but the user will be informed by an adequate and clear error message.

TIEMPO	INV	OR	SR	DISCR
0	200.00	0.00	0.00	0.00
1	200.00	0.00	0.00	0.00
2	200.00	0.00	0.00	0.00
3	200.00	0.00	0.00	0.00
4	200.00	0.00	20.00	0.00
5	180.00	10.00	20.00	20.00
6	170.00	15.00	20.00	30.00
7	165.00	17.50	20.00	35.00
8	162.50	18.75	20.00	37.50
9	161.25	19.38	20.00	38.75
10	160.63	19.69	20.00	39.38
11	160.31	19.84	20.00	39.69
12	160.16	19.92	20.00	39.84
13	160.08	19.96	20.00	39.92
14	160.04	19.98	20.00	39.96
15	160.02	19.99	20.00	39.98
16	160.01	20.00	20.00	39.99
17	160.00	20.00	20.00	40.00
18	160.00	20.00	20.00	40.00
19	160.00	20.00	20.00	40.00
20	160.00	20.00	20.00	40.00

Table 1 Inventory system tabulated results

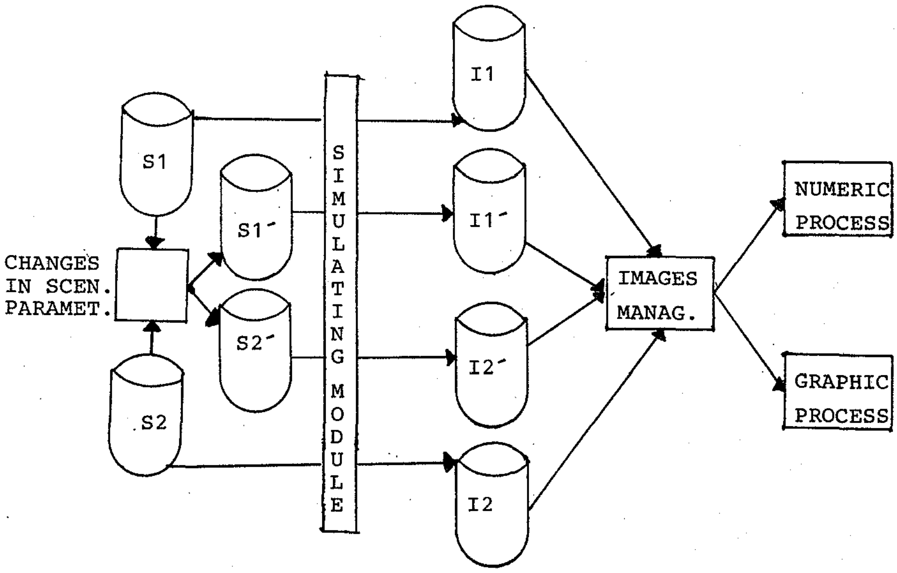


Fig. 5 Changes in parameters. Images Management

IMAGES

Once all the results derived from a run are stored in the disk as an image, we can operate over them to obtain some graphics, interesting reports or informations about the modeled system , and its behaviour along the time horizon

Images module permits to work around an image as well as com-- parate two different images variables and results, correspon-- ding to runs with different scenarios (Fig. 5).

We can use Graphics or Numerical processing sub-modules

Graphics

Using graphics sub-module, one can display the coloured curves describing the behaviour of the selected variables at this mo- ment, chosen among all the variables. They can be plotted at the same 0-1 scale as well as at the highest value variable / scale. 0-1 scale is useful to compare qualitative behaviour of the variables, and the highest value one is to compare quanti- tative results along the time horizon of the simulation. Figu- res 6a. and 6c. show both vertical scales options.

Bars diagrams are available too, as Figure 6b. shows.

An interesting feature is that plots may be plotting a varia-- ble in relation to the time or in relation to other variable / of the model.

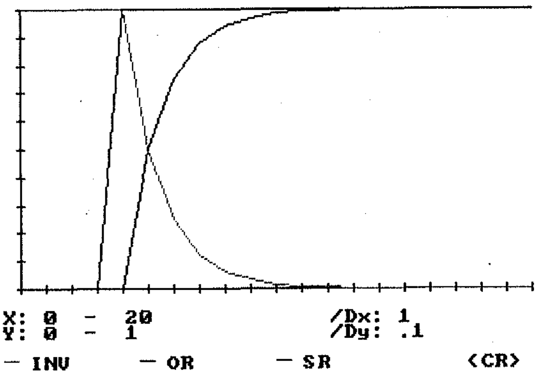


Fig. 6 (a) Inventory system plot, using a 0-1 scale

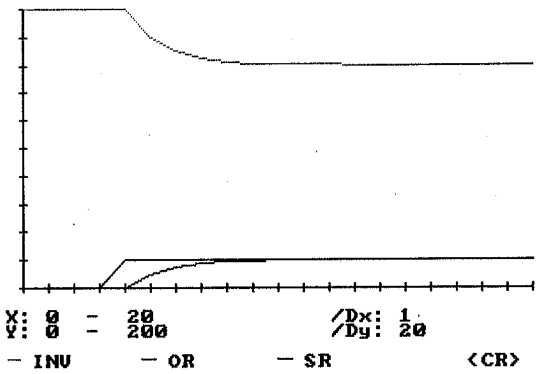


Fig. 6 (c) Inventory system plot, using the highest value scale

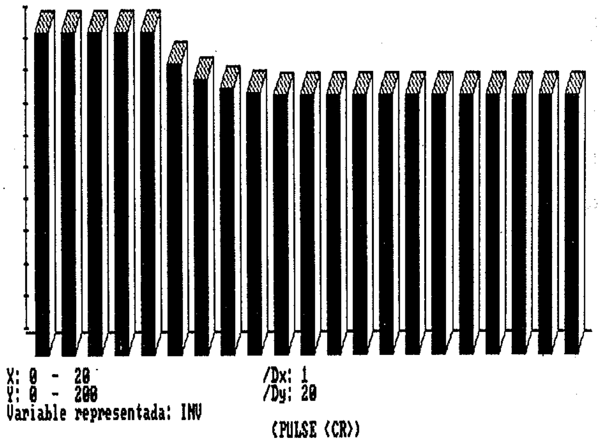


Fig. 6 (b) INV bar diagram

Numerical Processing

This option is not yet implemented, in the moment that we are writing this paper, althought it is a part of the SDSE design and we hope it will be soon, besides to some extensions of / the package as we will explain later.

The design includes possibilities as calculation of the minimum or maximum value of a variable along the run, arithmetic/ mean, mode, standard deviation and other interesting statis-- tics.

Individual values of the run, along the time, such as the INV value at the fifth week, are separately accesible .

A built-in programmed calculator is available, to help the user to operate and work, in an interactive way, with the / quantities and values that Numerical Processing module offers to him.

VISUAL AIDS LIBRARY AND ERROR MESSAGES

For each system input option, it will be possible to request/ to the system for visual aids about the possible allowed entries at this point, or about some of the symbols displayed / on the screen at that instant. In this way, the user never feels lost. Moreover, he can forget in a short time about the user´s handbook.

Anyway, if the user makes a mistake, such as not allowed input, scenario, model or variable identifier duplication, mispressing, wrong entry format, etc.., the system detects the mistake and displays an error message pointing out the kind of error. So, the user is able to correct the error by tiping again the input or returning to an adequate restart or rerun/ point.

Visual aids and Error messages are recorded on a file, what / makes easy to translate, add or change them.

CONCLUSIONS

SDSE takes in account the basic aspects which are more fre- / quently used along the model building and operating using / the systems dynamic techniques.

Educational world oriented, it has some limitations, which / may be important,from a professional viewpoint, to confront some kind of problems: the number of variables is limited , depending of their type, about 200 or 300 are available; Subroutines between the equations of the model, as being a part of it, cannot be defined; subscripted variables as elements/ of the model are not available.

In spite of these problems, we think that SDSE is an useful / skill, as it has been showed along Atenea project courses development in Murcia (ATEMUR courses), to be used in educative environments. We really hope that this will be, at least in some spanish teaching institutions, in a short period of time

However, we are convinced about the need for improving the package to offer a wider set of aids, messages and possibilities.

To finish, we can say that the most important feature that we think of the system is that it can be easily understood and used by teachers and pupils. We think that it provides a balance between ease of use and power, enough to confront the standard models in education.

FUTURE DEVELOPMENTS.

Nowadays, we are working to complete the Images module by im-- proving graphics and numerical processing in SDSE.

Further, some projects around SDSE as a sensitivity analysis / module and an automatic reports generator are in working.

Another interesting future project is the design and develop-- ment of an automatic bi-compiler to convert Microdynamo equa-- tions to SDSE equations and scenarios and vice versa, to try / to decrease the lost of portability due to the differences / between both systems.

In a different way, SDSP project (System Dynamics Software for Professional) is in the design phase althoght some modules / have already been implemented. The basis of SDSP is similar to that of SDSE, but SDSP is more powerful and commercial and industrial-world oriented. SDSP makes possible to use DS-LP mi-- xed models (18), (19) because a simplex algorithm is supplied, in a built-in way. Other features are: Runge-Kutta numerical / integration, different kinds of random distribution functions to choice, Pascal procedures and functions inclusion trough / the model, and subscripted or arrays as variables of the model equations.

SDSP is being implemented using TURBOPASCAL language, under MS DOS operating system for IBM PC and compatible.

ACKNOWLEDGEMENTS

We thank to SECOINSA Computers Company (now FUJITSU ESPAÑA) the financial aids, by means of grants for last courses computer students, who worked as programmers and testers in SDS project. SDS was the precursor of the actual SDSE, althoght the last cannot be considered as a release of the first because of the differences of design and internal structure. Anyway it was useful to constitute a work-group and to begin to work in projects of system dynamics software.

In spite of these problems, we think that SDSE is an useful skill as it has been showed along Atenea project courses, developed in Murcia (Spain) , to be used in educative environments. We really hope that this will be so, at least in some spanish teaching institution, in a short period of time.

However,we are convinced too about the need for improving the / package to offer a wider set of aids, messages and possibilities.

To finish, we can say that the most important feature that we / think of the system is that it can be easily understood and used by teachers and students. We think that it provides a balance / between ease of use and power, enough to confront the standard models in education.

FUTURE DEVELOPMENTS

Nowadays, we are working to complete the Images module by impro-- ving graphics and numerical processing, in SDSE.

Further, some projects around SDSE as a sensitivity analysis mo-- dule and an automatic reports generator are in working.

Another interesting future project is the design and development of an automatic bi-compiler to convert Microdynamo equations to SDSE equations and scenarios and vice versa, to try to decrease / the lost of portability due to the differences between both / systems.

On the other hand, SDSP project(System Dynamics Software for Professional) is in the design stage, althought some modules have already been implemented. The basis of SDSP are similar to those of SDSE, but SDSP is though to be more powerful and flexible. Commercial and industrial-world oriented, SDSP makes possible to use SD-LP mixed models (System Dynamics-Linear Programming), because the simplex algorithm is built-in.Toval(1985,1986). Other features are: Runge-Kutta numerical integration, different kinds of random distribution, Pascal procedures and functions to be included trough the models, and subscripted or arrays as variables / of the model equations.

SDSP is being implemented using TURBOPASCAL language, under MS / DOS operating system for IBM PC and compatible.

ACKNOWLEDGEMENTS

We thank to SECOINSA Computers Company (now FUJITSU ESPAÑA), the financial support, by means of grants for last courses computers students, who worked as programmers and testers in SDS project . SDS project was the precursor of the actual SDSE, althought the last cannot be considered as a release of the first because of / the deep differences of design and internal structure. Anyway it was very useful to constitute a work-group and to begin to work/ around system dynamics software projects

REFERENCES

Atenea ,Proyecto (1985). Ministerio de Educación y Ciencia de España. Documento nº 1.

Balkovich, E. (1985). Computing in hicher Education: The Athena experience. Communications of the ACM, Nov.1985 , vol.28,nº 11.

Breuer, K. (1985). Computer simulation and cognitive development. WCCE85 Computers in Education. Elsevier Science Publishers B.V. (North Holland) IFIP 1985

Eyferth, K.(1974). Computer im Unterricht (Klett, Stugar , 1974).

Golden,D.G.(1985). Software Engineering considerations for / the design of simulation languages. Simulation, October 1985. vol.

Toval, A. (1985). Uso de técnicas de programación lineal en Dinámica de Sistemas. Actas del VI Congreso de Informática y Automática. Servicio de Publicaciones E.T.S. I. de Telecomunicaciones. Ciudad Universitaria s/n , (28040).Madrid

Toval, A. (1986) Murcia A/I, A mixed system dynamics and linear programming model for regional investments planning. Proceedings of the 1986 Int.Conf. of the System Dynamics Society. Univ. of Sevilla. Spain.

Zinn,K.L. (1973) An evaluative review of uses of computers in ins- / truction, Project CLUE final report. Univ. of Michigan, 3rd edition 1973