TEACHING AND RESEARCH NEWS

The Approach to System Dynamics at Griffith University, Queensland, Australia *by W.E. Cundiff*

1. AN INTEGRATED APPROACH

In the School of Social and Industrial Administration at Griffith University, students concentrating in *"Decision Analysis and Support Systems"* undertake a defining course in Simulation and Modelling. This course provides third year students with a theoretical and practical orientation for subsequent, more advanced management science course work or further development of simulation in particular. The first course aims to convey the scope and breadth of the field while at the same time leaving the student with some in-depth skill and appreciation of simulation. The course is divided into two parts focussing each on the familiar continuous and discrete approaches.

Students in their second year undertake an intensive study of structured programming using Pascal. This had the advantage of maintaining continuity and developing further the student's skill in structured programming. Furthermore, it was decided to utilise the on-campus Apple laboratory rather than the PDP-10 of the University of Queensland. This had the added advantage of reducing overhead in that Girffith University purchases time on the PDP-10 via a *"real money"* charging arrangement. The decision was further influenced by the current heightened interest in the simulation field regarding micro-computer applications and, in particular, strongly typed languages such as Pascal, Ada and Simula.[1,2]

The discrete-event component is based around micro-PASSIM, a GPSS-like set of block procedures originally developed at the University of British Columbia and later enhanced to run on the Apple computer by Claude C Barnett of Walla Walla College in the United States.[3,4] The continuous systems component centres on the use of PASCSL (PAScal-based Continuous Simulation Library) developed in the School of Social and Industrial Administration at Griffith University. Both packages are implemented as separately compiled units and are resident in libraries on the Apple Pascal system.[5] PASCSL, an extension of earlier work in applying general purpose languages to System Dynamics, provides a range of functions and procedures for developing models consistent in large part with the DYNAMO language.[6,7,8] Numerous "housekeeping" operations are also handled in PASCSL such as type declarations, setting constants and global variables and file I/O. The user defines the model in Pascal, incorporating the various functions and procedures as required. The program is structured according to conventions applicable to virtually all models defined using the package. An added feature is the significantly faster model compilation time using externally

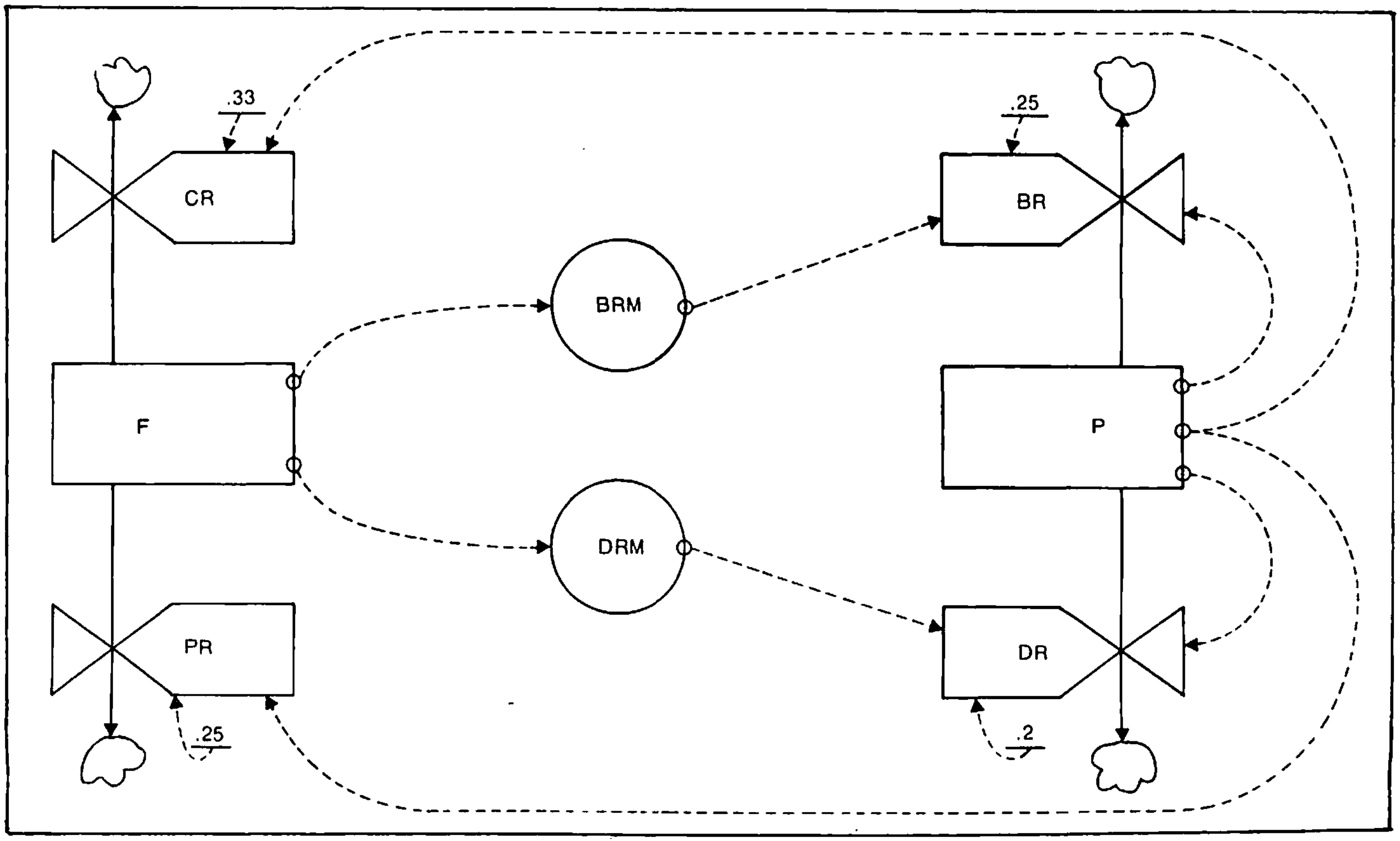

Figure 1: System Dynamics Flow Diagram

```
PROGRAM POP;
USES TRANSCEND,PASCSL;

VAR BRFMT,DRFMT:BIG;
    BRN,DRN,PRN,CRN,CR,PR,BR,DR,P,F,BRM,DRM:REAL;

BEGIN
INN(BRN,0,0);INN(DRN,0,1);INN(PRN,0,0);
INN(CRN,0,0);INN(P,0,0);INN(F,0,0);
READTABLE(DRFMT,1);READTABLE(BRFMT,0);

WHILE TIME<=LENGTH DO
  BEGIN
  CR:=P*CRN;PR:=P*PRN;
  BRM:=TABLE(BRFMT,F);DRM:=TABLE(DRFMT,F);
  BR:=BRN*P*BRM;DR:=DRN*P*DRM;
  P:=P+DT*(BR-DR);F:=F-DT*(CR+PR);
  WRITELN;WRITE(ROUND(TIME):4,BR:8:2,DR:8:2,P:9:2,F:9:2);
  RUN('    BUGS IN A JAM JAR');
  END;
END(*POP*).
```

Listing 1:

compiled units. Also, because of file handling, a model need not be recompiled when assumptions have changed by assigning new values to parameters. Data may be stored in an input file and edited prior to running a simulation. The package also promotes efficient memory management in that the units are segmented and swapped in and out at run time. Swapping is of prime importance when running large and complex models.

2. DATA AND ALGORITHMS

Listing 1 is an example of a Pascal program that embodies a generic model of carrying capacity. The model as conceived in the flow diagram (Figure 1) and programmed in DYNAMO (Listing 2) is based on the assumption that a Fly population (P) is situated within a closed environment (a bugs in a jam jar experiment) containing a finite supply of food (F). The quality and adequacy of the food influences both their birth rate (BR) and death rate (DR). A top-down discussion will outline the use of the package and the coding conventions. Initially the program must designate that it *uses* two units, the transcendental function unit and PASCSL, both resident in the System Library. Variables are then declared with any tables declared to be of type BIG. The type declaration for BIG is done behind the scenes by PASCSL; the user need not ever make type declarations. After the variable declarations the main body of the program will first contain the statements for reading the initial values for parameters and variables driving the model.

The procedure INN is provided for reading scalars while READTABLE is supplied for reading vectors that contain the table co-ordinates. Both have parameters for controlling the display of data at run time (a 1 in the far right signals display) while INN also permits rounding and formatting depending on the value of the middle parameter. Table 1 represents the input file that also follows a convention for model building with PASCSL. The topmost number is the initial value for the global TIME, the starting period of the simulation. The second value is LENGTH which specifies the number of periods that the simulation will run. The third scalar is DT to be used in setting the size of the interval in integrating the model equations; this is done using an algorithm for rectangular approximation of the Euler form. Note that none of the three values, TIME, LENGTH and DT, are read within the program POP. All models will require these values and they are read automatically by a PASCSL routine. The six values in line four are the birth rate norm (BRN), death rate norm (DRN), pollution rate norm (PRN), consumption rate norm (CRN), initial population size (P) and initial units of food (F). These are each read with INN.

Lines five and six contain the values for the death rate from food multiplier table and the birth rate from food multiplier table respectively. As with DYNAMO, the values for

```
* POP
L P.K=P.J+(DT)(BR.JK-DR.JK)
N P=PI
C PI=10
R BR.K=(.25)(P.K)(BRM.K)
A BRM.K=TABLE(BRMT,N.K,0,15E3,3E3)
T BRMT=0/.1/.75/1/3/5
R DR.KL=(.2)(P.K)(DRM.K)
A DRM.K=TABLE(DRMT,N.K,0,15E3,3E3)
T DRMT=4/3/1/.5/.1/.1
L F.K=F.J-(DT)(CR.JK+PR.JK)
N F=FI
C FI=10000
R CR.KL=(P.K)(.33)
R PR.KL=(P.K)(.25)
SPEC DT=1/LENGTH=50/PRTPER=0/PLTPER=1
PLOT BR=B/DR=D/P=P/F=F
RUN
```

Listing 2:

```
1.0
50.0
1.0
0.25 0.20 0.25 0.33 10.0 10000.0
0.0 15000.0 3000.0 4.0 3.0 1.0 0.5 0.1 0.1
0.0 15000.0 3000.0 0.0 0.1 0.75 1.0 3.0 5.0
```

Table 1:

the independent variable are entered as the lower and upper values for the range (0.0 – 15000.0) and the increment of increase (3000.0). The remaining six values are the Y coordinates.

The next PASCSL programming convention is the WHILE construct as shown with the logical test on the global TIME and LENGTH. The compound statement in the loop contains the actual model equations including the computations of the multiplier coefficients, BRM and DRM using the procedure TABLE. Standard Pascal statements are used for writing to an output file or to the screen; Table 2 shows the results of a model run. PASCSL controls the increment to the counter and displays the message parameter in RUN upon completion. As with DYNAMO, the system will issue an error message if an auxiliary falls outside the range of its independent variable. PASCSL also supports TABHL which sets the multiplier coefficient to the Y value corresponding to the high/low extremes in the range of the independent variable, in the event of a value generated outside the designated range for the auxiliary. Other DYNAMO functions supported include DELAY1, DELAY3, CLIP, SWITCH, MIN, MAX and EXP.

Once a model has been defined around the appropriate procedures, it is first compiled; then after compilation, the externally compiled units must be linked into the host program (e.g. POP). The model can then be simulated.

CONCLUSION

Within the Griffith/SIA course, such models, from the simple to detailed, have been programmed to study wide, interdisciplinary areas such as:

- Effects of interest rate growth and exponential decay
- Thermostatic behaviour
- Immigration and urban population growth
- Effect of immunity loss in the epidemic spread of influenza
- Micro-economic supply/demand of manufactured goods
- Economic feasibility of extraction of finite mineral reserves

Students, competent in Pascal, upon entering the course, find the continuity of approach very helpful in gaining an understanding of the substantive side of modelling. Technical concepts such as functions, procedures, modular design, stepwise refinement and data structures are brought to bear on the content of a modelling problem. Concepts shared between discrete and continuous approaches are communicated easily and from experience, students may undertake a modelling task at greater depth more quickly. This is of particular importance given the wide scope of material to be covered within the short time of a single semester.

TIME	BIRTHS	DEATHS	FLIES	FOOD
1	4.17	0.73	13.43	9994.20
2	5.58	0.99	18.03	9986.41
3	7.47	1.33	24.17	9975.95
4	9.98	1.79	32.36	9961.93
5	13.28	2.41	43.23	9943.16
6	17.60	3.24	57.60	9918.09
7	23.21	4.35	76.47	9884.68
8	30.39	5.84	101.01	9840.33
9	39.40	7.84	132.58	9781.74
10	50.42	10.49	172.50	9704.85
11	63.39	14.01	221.88	9604.80
12	77.84	18.61	281.11	9476.10
13	92.58	24.54	349.15	9313.06
14	105.50	32.00	422.65	9110.55
15	113.45	41.02	495.09	8865.41
16	122.38	51.73	565.74	8578.26
17	136.46	64.53	637.68	8250.14
18	149.46	79.71	707.43	7880.28
19	160.35	97.15	770.64	7469.98
20	168.09	116.37	822.36	7023.01
21	171.72	136.43	857.65	6546.04
22	170.57	155.92	872.30	6048.60
23	164.44	173.05	863.69	5542.66
24	140.55	225.40	778.83	5041.72
25	105.60	255.28	629.16	4590.00
26	69.92	244.11	454.96	4225.09
27	41.56	198.66	297.86	3961.21
28	22.96	140.54	180.28	3788.45
29	12.21	89.21	103.27	3683.89
30	6.41	52.54	57.13	3623.99
31	3.36	29.53	30.96	3590.86
32	1.77	16.14	16.59	3572.90
33	0.93	8.69	8.83	3563.27
34	0.49	4.64	4.69	3558.15
35	0.26	2.46	2.48	3555.43
36	0.14	1.31	1.31	3553.99
37	0.07	0.69	0.69	3553.23
38	0.04	0.37	0.37	3552.83
39	0.02	0.19	0.19	3552.61
40	0.01	0.10	0.10	3552.50
41	0.01	0.05	0.05	3552.44
42	0.00	0.03	0.03	3552.41
43	0.00	0.02	0.02	3552.39
44	0.00	0.01	0.01	3552.39
45	0.00	0.00	0.00	3552.38
46	0.00	0.00	0.00	3552.38
47	0.00	0.00	0.00	3552.38
48	0.00	0.00	0.00	3552.38
49	0.00	0.00	0.00	3552.38
50	0.00	0.00	0.00	3552.38

Table 2:

REFERENCES

1. Society for Computer Simulation, *"Call for Papers: 1984 SCS Multiconference on Simulation in Health Care Delivery Systems, Modelling and Simulation on Microcomputers, and Simulation in Strongly Typed Languages ADA, PASCAL, SIMULA, ...",* SCS 1983 – Complete Publications Listing, *La Jolla, 1983.*

2. U.K. Operational Research Society, *"Special Issue: Management, Operational Research and the Micro",* Journal of the Operational Research Society, Vol. 32, No. 4, April, 1981.

3. UYENO, Dean H., *"PASSIM: A Discrete-Event Simulation Package for Pascal",* Simulation, December, 1980, pp.183-190.

4. BARNETT, Claude C., *"Micro PASSIM: A Discrete-Event Simulation Package for a Microcomputer Using UCSD Pascal",* Modelling and Simulation on Microcomputers, *La Jolla: Society for Computer Simulation, 1982.*

5. TONKENS, R.M., *"A Guided Tour of Apple Pascal Units and Libraries",* BYTE, Vol. 7, No. 2, February 1982, pp. 225-244.

6. CUNDIFF, W.E., *"Simulating System Dynamics in APL",* Applied Mathematical Modelling, Vol. 3, August 1979, pp. 312-314.

7. CUNDIFF, W.E., *"An APL Function for the Approximation of Multiplier Values in System Dynamics Models",* Applied Mathematical Modelling, Vol. 5, June 1981, pp. 212.

8. CUNDIFF, W.E., *"Simulating Complex Systems – Personally",* Occasional Paper No. 36, Griffith University, School of Social and Industrial Administration, July 1983.