

Kruger, P. S.; Lindeque, P.; Adendorff, K.

Article

Simulation and the simulation language SLAM II as a systems analysis tool

South African Journal of Business Management

Provided in Cooperation with:

University of Stellenbosch Business School (USB), Bellville, South Africa

Suggested Citation: Kruger, P. S.; Lindeque, P.; Adendorff, K. (1989) : Simulation and the simulation language SLAM II as a systems analysis tool, South African Journal of Business Management, ISSN 2078-5976, African Online Scientific Information Systems (AOSIS), Cape Town, Vol. 20, Iss. 3, pp. 154-162,
<https://doi.org/10.4102/sajbm.v20i3.955>

This Version is available at:

<https://hdl.handle.net/10419/218031>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<https://creativecommons.org/licenses/by/4.0/>

Simulation and the simulation language SLAM II as a systems analysis tool

P.S. Kruger, P. Lindeque and K. Adendorff

Centre for Computer Integrated Production Systems, Department of Industrial and Systems Engineering, University of Pretoria, Pretoria, 0002 Republic of South Africa

Accepted 24 November 1988

This paper provides a general introduction to the simulation technique and the simulation language SLAM II. The use of simulation modelling and in particular SLAM II is demonstrated by way of an illustrative example consisting of the building and manipulation of a simulation model of a materials handling subsystem of a beer brewing and packaging plant. The purpose of the model is to investigate the influence of the number of loading and unloading docks on various measures of system performance.

Hierdie artikel verskaf 'n inleiding tot die simulasietegniek en die simulasietaal SLAM II. Die gebruik van simulasiemodellering en spesifiek SLAM II word gedemonstreer met behulp van 'n illustratiewe voorbeeld bestaande uit die ontwikkeling en manipulasie van 'n simulasiemodel van 'n materiaalhanteringsstelsel van 'n bierbrouery en verpakkingsaanleg. Die doelwit van die model bestaan uit 'n ondersoek na die invloed van die aantal laai- en aflaafasileite op verskeie maatstawwe van stelselleffektiwiteit.

Introduction

Technological development and the ever-increasing demand for goods and services in an industrialized society have resulted in the adoption of manufacturing, commercial and management systems which continue to grow in size and complexity. At the same time political, social, and economic systems are becoming less stable and therefore probably less predictable, resulting in the fact that poor decisions are becoming more costly. The necessity for effective and efficient system design and decision making is therefore apparent as is the need for procedures and techniques to support the systems analyst.

A number of such techniques have been developed, some of which have been and are still being used with excellent results. Most of these techniques rely in one way or another on the concept of building a model of the system under investigation.

Surveys have indicated that simulation is the most widely used quantitative modelling technique employed as a systems analysis tool by industry and government in the United States and this may become true for South Africa in the near future. Simulation is without doubt one of the most powerful techniques available for improving productivity through effective analysis, design and operation of complex processes and systems. The concept of simulation is both simple and intuitively appealing to managers, engineers, systems analysts or anyone involved in decision making. Simulation has often been described as a course of last resort in relation to other modelling techniques but this may no longer be true. The increasing availability of relatively low cost computer hardware and especially the development of specialized, easy to use, but powerful and flexible simulation languages have enhanced the feasibility of using simulation over other modelling techniques to such an extent that it may be the preferred approach in the majority of instances.

A number of different simulation languages and simulation support software for use on micro-computers

and main frame computers are available (Catalog, 1983). In the opinion of the authors the simulation language SLAM II is one of the more versatile, powerful, and better supported languages available.

THE SLAM II simulation language

Since its introduction in 1979 (Pegden & Pritsker, 1979) SLAM has been widely acclaimed as the state-of-the-art in simulation languages and has been subjected to a continuing process of development and enhancement in terms of capabilities and scope. Since inception it has been used to build simulation models of a wide variety of processes and systems (Pritsker, 1982). SLAM II, version 4, is an advanced FORTRAN-based Simulation Language that allows alternative approaches to modelling. Requiring only a FORTRAN compiler, it is easily implemented and maintained on, or transported between, different computer systems and, if necessary, it can easily be modified or extended for particular applications. It allows the analyst to develop models from a process interaction, next event, or continuous perspective, by combining network, discrete event, and continuous modelling capabilities and explicitly defines the interfaces between the modelling approaches to allow new conceptual views of systems to be explored.

The SLAM II simulation language consists of three separate, but interlocking subsystems.

The first subsystem, representing the PROCESS INTERACTION world view, consists of a number of branches linking graphical symbols called nodes. Each node and/or branch represents an often-encountered process in general terms such as a queue or an activity. Using these nodes and branches it is often possible to build a graphical network model of a system displaying the constituent processes of the system as well as the interaction between processes and the logical flow of discrete entities through the system. The network part of SLAM II is probably the easiest, fastest and most often used way of building a simulation model and is in many ways similar to other well known discrete simulation

languages like GPSS (Gordon, 1975). Each node and its parameter values are described by an input statement and these input statements together with certain control statements form a program that will be interpreted and executed by a computer running the SLAM II software.

The second subsystem of SLAM II, representing the NEXT EVENT world view, consists of a number of user-callable FORTRAN support routines. Each of these routines performs a basic, often repeated function necessary in next event oriented simulation models such as random number generation, event scheduling, and file manipulation. The user may use these routines together with user-generated code, for example event logic, to construct a simulation model, coded in FORTRAN, of the system under consideration. Although this approach is somewhat more difficult to use than the network approach and requires detailed knowledge of the FORTRAN programming language, the inherent flexibility is almost unlimited.

The third subsystem of SLAM II, representing the CONTINUOUS world view, also consists of a number of user-callable FORTRAN support routines. Using these routines, together with user-generated FORTRAN code a continuous simulation model, which often consists of a system of differential or difference equations, may be built and manipulated quite easily and effectively.

Each of the three subsystems discussed, explicitly contains and defines the necessary interfaces between the three modelling approaches, enabling the user to use any combination of approaches within the same model. This facility makes a particular important contribution to the flexibility and modelling capability of SLAM II. The illustrative example described later will use both the process interaction and the next event approaches.

In all three subsystems of SLAM II adequate

provision has been made to control the simulation run, collect data concerning the values of systems variables and for displaying and summarizing the data in numerical and graphical reports.

SLAM II is only part of a family of available simulation support software. These separate, but fully compatible, software include such features as enhanced graphical analysis and output, animation, statistical analysis of simulation output data, parameter estimation, special materials handling capabilities, and data base manipulations.

Being a FORTRAN based language, it is possible when using SLAM II to modify or extend the software for a particular application if necessary. It is for example relatively easy to generate model-oriented or user-specified output reports.

According to Pritsker & Pegden (1979), simulation models may be employed at four different levels:

- as explanatory devices to define a system or problem;
- as analysis vehicles to determine critical elements, components and issues;
- as design assessors to synthesize and evaluate proposed solutions, and
- as predictors to forecast and aid in planning future developments.

The SLAM II simulation language is an effective tool in the development of simulation models that will be employed at anyone of these application levels. It is well documented, well supported and probably most importantly has been and will be subjected to a continuous process of development and enhancement (Pritsker & Pegden, 1979).

Simulation languages in general, and therefore also SLAM II, are not without disadvantages. In our judgement, the most important disadvantage lies in the

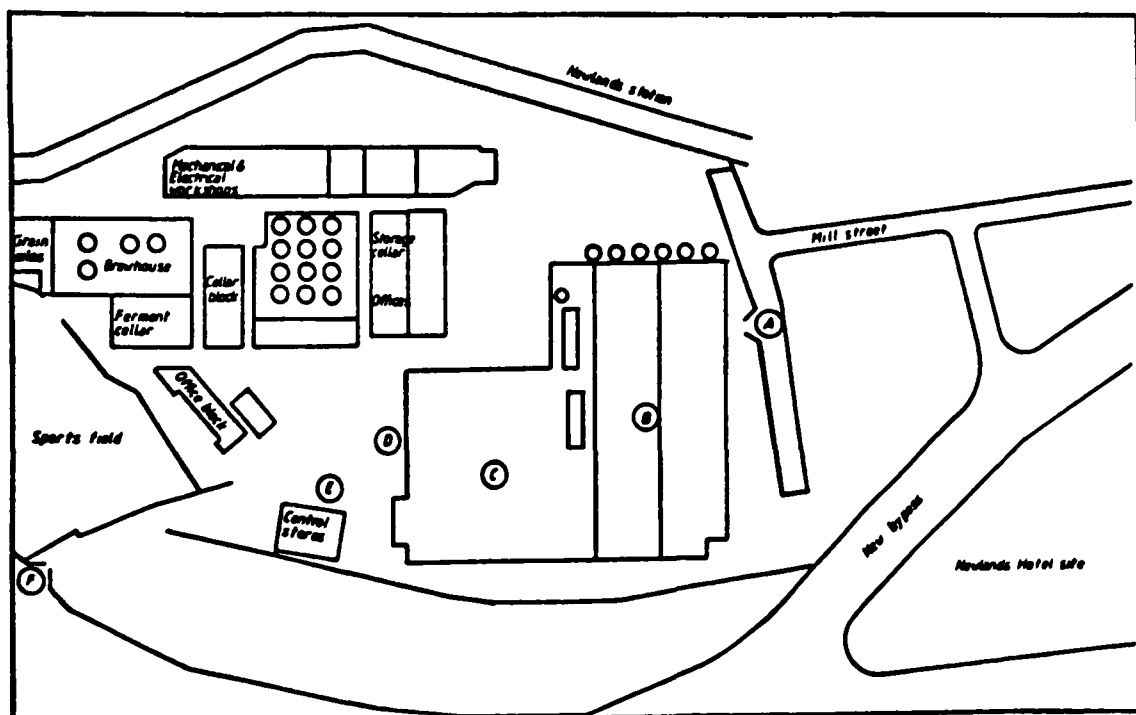


Figure 1 Schematic layout of the beer brewing and packaging materials handling system

GEN,P. LINDEQUE,SAB,05/16/84,1;
LIMITS,12,2,500;
NETWORK;

;
;
;

*****RESOURCE BLOCKS*****

RESOURCE/DOCK(10),3;
RESOURCE/FLT(6),7,8;
RESOURCE/TRA(0),11;
RESOURCE/TRB(0),12;

DOCK RESOURCE.
FORKLIFT TRUCK POOL.
DUMMY RESOURCE.
DUMMY RESOURCE.

;
;
;

*****TRUCKS ARRIVE AND QUEUE AT GATE*****

CREATE,TRIAG(20,21,29),,1;
ASSIGN,ATRIB(2)=11;
ACT,,NA;
CREATE,TRIAG(46,48,65),,1;
ASSIGN,ATRIB(2)=12;
ACT,,NA;
NA QUEUE(1);
ACT,TRIAG(1.3,1.5,2.5);
EVENT,1;
TERM;

GENERATE ARRIVAL OF
PRODUCT #1 TRUCKS.

GENERATE ARRIVAL OF
PRODUCT #2 TRUCKS.

QUEUE AT FIRST GATE.
SERVICE ACTIVITY AT GATE.
CALL EVENT ROUTINE.

;
;
;

*****DOCKING AND LOADING OF TRUCKS*****

ENTRY,1;
AWAIT(3),DOCK/1;
GOON,1;
ACT,,0.2,NB;
ACT,,0.8;
AWAIT(9),FLT/1;
ACT,TRIAG(25,28,36);
FREE,FLT/1;
ACT,,NB;

ENTER FROM EVENT ROUTINE.
AWAIT A DOCK RESOURCE.

BRANCH TRUCKS NOT UNLOAD-
ING (20%) TO NODE NB.
AWAIT A FORKLIFT TRUCK.
UNLOAD TRUCK.
FREE FORKLIFT TRUCK.

NB GOON,1;
ACT,,ATRIB(2).EQ.11,NC;
ACT,,ATRIB(2).EQ.12,ND;
NC ALTER,TRA/22;
QUEUE(4);
ACT,REL(PRA),,NE;
ND ALTER,TRB/22;
QUEUE(5);
ACT,REL(PRB),,NE;
NE EVENT,2;
QUEUE(6);
ACT,TRIAG(1.3,1.5,2.5);
COLCT,INT(1),TIME IN SYSTEM;
TERM;

BRANCH TRUCKS TO CORRECT
PRODUCT LINE.
DUMMY RESOURCE PROD #1.
AWAIT COMPLETION OF
LOADING ACTIVITY.
DUMMY RESOURCE PROD #2.
AWAIT COMPLETION OF
LOADING ACTIVITY.
CALL EVENT ROUTINE.
QUEUE AT SECOND GATE.
SERVICE ACTIVITY AT GATE.
COLLECT STATISTICS.

;
;
;

*****PRODUCT LINE#1*****

CREATE,TRIAG(.90,.95,1.30);
ACT,,NF;
ACT,,NG;
NF ACCUM,22,22;
ASSIGN,ATRIB(2)=11;
EVENT,3;
TERM;
NG AWAIT(11),TRA/1;
AWAIT(7),FLT/1;
ACT,,5;
FREE,FLT/1;
PRA ACCUM,22,22;
FREE,TRA/22;
ALTER,TRA/-22;
TERM;

CREATE PRODUCT #1.

ALERT WAITING TRUCKS THAT
22 UNITS OF PRODUCT #1
HAS ACCUMULATED.

AWAIT DOCKING OF TRUCK.
AWAIT FORKLIFT TRUCK.
LOADING ACTIVITY.
FREE FORKLIFT TRUCK.
22 UNITS PER TRUCK.
FREE DUMMY RESOURCE.
REMOVE DUMMY RESOURCE.

Figure 3 The SLAM input statements for the network model

```

;
;
      CREAT,TRIAG(2.1,2.2,3.0);
      ACT,..NH;
      ACT,..NI;
NH  ACCUM,22,22;
      ASSIGN,ATRI(2)=12;
      EVENT,3;
      TERM;
NI  AWAIT(12),TRB/1;
      AWAIT(8),FLT/1;
      ACT,..5;
      FREE,FLT/1;
PRB ACCUM,22,22;
      FREE,TRB/22;
      ALTER,TRB/-22;
      TERM;
      ENDNETWORK;
INIT,0,1561;
MONTR,CLEAR,120;
MONTR,TRACE,0,10;
FIN;

```

*****PRODUCT LINE #2*****

```

      CREATE PRODUCT #2.

      ALERT WAITING TRUCKS THAT
      22 UNITS OF PRODUCT #2
      HAS ACCUMULATED.

      AWAIT DOCKING OF TRUCK.
      AWAIT FORKLIFT TRUCK.
      LOADING ACTIVITY.
      FREE FORKLIFT TRUCK.
      22 UNITS PER TRUCK.
      FREE DUMMY RESOURCE.
      REMOVE DUMMY RESOURCE.

      SET SIMULATION RUN LENGTH
      CLEAR STATISTICS.
      PRINT TRACE REPORT.

```

Figure 3 Continued

```

C *****SUBROUTINE EVENT*****
C
      SUBROUTINE EVENT(I)
      DIMENSIONATR(100)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
      1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      IAT = IFIX(ATRI(2))
      GOTO (100,200,300) I
C
C *****EVENT(1)*****
C
C IF MORE THAN 22 UNITS OF THE PRODUCT AND A DOCK IS AVAILABLE THE
C TRUCK IS PUT INTO THE NETWORK ELSE THE TRUCK IS PUT IN FILE 2.
C
100 IF (NNQ(IAT).GE.22.AND.NNRSC(1).GT.0) THEN
      CALL ENTER (1,ATRI)
      ELSE
      CALL FILEM(2,ATRI)
      ENDIF
      RETURN
      ;PUT TRUCK IN NETWORK.
      ;FILE THE ENTITY.
C
C *****EVENT(2)*****
C
C SCAN FILE 2 AND PUT THE FIRST TRUCK FOR WHICH 22 PRODUCT
C UNITS EXIST INTO THE NETWORK.
C
200 CALL FREE(1,1)
      DO210 ICOUNT = 1,NNQ(2)
      CALL COPY (ICOUNT,2,ATR)
      IATR = IFIX(ATRI(2))
      IF (NNQ(IATR).LT.22)GOTO 210
      CALL RMOVE(ICOUNT,2,ATR)
      CALL ENTER(1,ATR)
      RETURN
      ;FREE A DOCK RESOURCE.
      ;SCAN FILE 2 FROM TOP.
      ;COPY ATTRIBUTES OF ENTRY.
      ;INTEGER VALUE OF ATR.
      ;TEST FOR PRODUCT UNITS.
      ;REMOVE ENTRY FROM FILE.
      ;PUT TRUCK IN NETWORK.
210 CONTINUE
      RETURN
C
C *****EVENT(3)*****
C
C IF A DOCK RESOURCE IS AVAILABLE SCAN FILE 2 AND PUT THE FIRST
C TRUCK OF TYPE TEST INTO THE NETWORK.
C

```

Figure 4 The FORTRAN code for the next event sub model

```

300  IF (NNRSC(1).LE0) RETURN
      TEST = ATRIB(2)
      DO 310 ICOUNT = 1, NNO(2)
      CALL COPY(ICOUNT,2,ATR)
      IF (ATR(2).NE.TEST) GOTO 310
      CALL RMOVE (ICOUNT,2,ATR)
      CALL ENTER(1,ATR)
      RETURN
310  CONTINUE
      RETURN
      END

```

```

;TEST FOR DOCK RESOURCE.
;SCAN FILE 2 FROM TOP.
;COPY ATTRIBUTES OF ENTRY.
;TEST FOR TRUCK TYPE.
;REMOVE ENTRY FROM FILE.
;PUT TRUCK IN NETWORK.

```

Figure 4 Continued**Table 1** Assumptions and system parameter values

Description	Value
Number of docks	10
Number of forklift trucks	6
Time between arrivals of product #1 trucks (triangularly distributed between 20 and 29 with a modal value of 21)	TRIAG(20,21,29) min
Time between arrivals of product @2 trucks (triangularly distributed between 46 and 65 with a modal value of 48)	TRIAG(46,48,65) min
Service time at gates (triangularly distributed between 1,3 and 2,5 with a modal value of 1,5)	TRIAG(1.3,1.5,2.5) min
Percentage of trucks requiring unloading.	80%
Time for unloading activity (triangularly distributed between 25 and 36 with a modal value of 28)	TRIAG(25,28,36) min
Time between creations of product #1 (triangularly distributed between 0,90 and 1,30 with a modal value of 0,95)	TRIAG(.90,.95,1.30) min
Time between creations of product #2 (triangularly distributed between 2,1 and 3,0 with a modal value of 2,2)	TRIAG(2.1,2.2,3.0) min
Loading time per unit.	0,5 min
Maximum number of attributes per entity.	12
Number of user files.	2
Maximum number of concurrent entities in network.	500
Simulation runlength	20 000 min
Statistics cleared at time	120 min
SLAM trace report between time	0 and 10 min

the amount of in-process-inventory.

A secondary purpose would be to determine, for a given number of loading/unloading docks, the influence of the size of the forklift truck fleet on delays experienced by the trucks caused by the non-availability of forklift trucks.

The SLAM II model

The SLAM II simulation model consists of two interrelated submodels. The first submodel uses the process interaction approach (network model) and the second submodel uses the next event approach.

Network model description

The network model is shown in Figure 2 using the

standard SLAM node symbols. The relevant input statements of the network model is shown in Figure 3.

Next event model description

Figure 4 shows the FORTRAN code for the next event sub model.

Model inputs and assumptions

After formulating the model as described above, it is necessary to obtain the relevant data in order to make assumptions regarding the probability distributions and system parameter values that will be used. Performing this phase only after the preliminary model formulation has been completed ensures that only system data which are really necessary will be obtained.

Table 1 summarizes the assumptions and system parameter values that will be used.

At the time of performing the modelling exercise very little data were available with respect to system parameters such as time between arrivals, service times, loading/unloading times, and production rates. This was mainly due to the fact that it was a new proposed system. Furthermore, the purpose of this paper is primarily to illustrate the possible application of simulation modelling and it was therefore deemed adequate to use triangular distributions for these parameters. With more data available the appropriate distributions may be fitted and adapting the model to reflect these changes is very easy.

Model validation

Since the system under consideration is only a proposed, and therefore a non-existing system, it is not possible to perform model validation by comparing model outputs to observable outputs of the real system. It is, however, possible to obtain a trace when running a SLAM II model, depicting in detail what has happened to each entity in the system. Using the trace report the model executing logic may be checked and for the purposes of this paper this procedure will be considered as adequate model validation.

Model manipulation and results

The model was manipulated using an IBM compatible micro-computer with 640 Kbytes of random access memory, a 20 Mbyte hard disk, SLAM II and the MICROSOFT FORTRAN 77 compiler.

Figure 5 shows part of a typical SLAM summary report.

With reference to the SLAM summary report (Figure 5) the most important results of this specific simulation run may be summarized as follows:

1. The average time spent in the system by the trucks is 73,25 minutes with a minimum of 5,51 and a maximum of 139,6 minutes.
2. The average number of trucks waiting in line is 1,68 with a maximum of 4 trucks.
3. The average in-process inventory for product number one is 4,43 units with a maximum of 65 units.
4. The average in-process-inventory for product number two is 3,95 units with a maximum of 57 units.
5. The average utilization of the docks is 26,1%.
6. The average utilization of the fork-lift truck fleet is 35,3%.

By repeating the simulation experiment for various values of the number of loading/unloading docks available the results shown in Table 2 may be obtained.

By repeating the simulation experiments for various

SIMULATION PROJECT SAB

DATE 4/16/1985

CURRENT TIME .2000E+05

STATISTICAL ARRAYS CLEARED AT TIME .1200E+03

BY P. LINDEQUE
RUN NUMBER 1 OF 1

STATISTICS FOR VARIABLES BASED ON OBSERVATION

MEAN	STANDARD VALUE	COEFF. OF DEVIATION	MINIMUM VARIATION	MAXIMUM VALUE	NUMBER VALUE	OBSER.
TIME IN SYSTEM	.7325E=02	.2740E+02	.3741E+00	.5510E=01	.1396E+03	1223

FILE STATISTICS

FILE NUMBER	ASSOCIATED NODE TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAITING TIME
1	QUEUE	.0027	.0521	1	0	.0441
2	1.6822	.9902	4	3	28.5102	
3	AWAIT	.0000	.0000	1	0	.0000
4	QUEUE	.0801	.2726	2	0	1.8733
5	QUEUE	.0020	.0444	1	0	.1049
6	QUEUE	.0023	.0477	1	0	.0371
7	AWAIT	.2520	1.6012	22	0	.2673
8	AWAIT	.1255	1.0373	21	0	.3019
9	AWAIT	.0004	.0190	1	0	.0075
10	.0000	.0000	0	0	.0000	
11	AWAIT	4.1799	6.7430	43	0	4.4339
12	AWAIT	3.8283	6.0012	36	0	9.2071
13	6.3386	1.4286	33	7	.5920	

RESOURCE STATISTICS

RESOURCE NUMBER	RESOURCE LABEL	CURRENT CAPACITY	AVERAGE UTILIZATION	STANDARD DEVIATION	MAXIMUM UTILIZATION	CURRENT UTILIZATION
1	DOCK	10	2.6125	.6488	5	3
2	FLT	6	2.1205	1.3279	6	3
3	TRA	22	9.0362	7.1626	32	19
4	TRB	22	7.7856	8.1899	37	16

Figure 5 SLAM summary report

Table 2 The influence of the number of docks

		Number of docks			
		>4	4	3	2
Time in system	Mean	73,25	88,46	246,1	1326
	Min	5,51	5,51	4,68	4,89
	Max	139,6	209,9	567,4	3983
Number of trucks in E	Mean	1,68	2,76	13,04	85,44
	Max	4	7	22	168
In process inventory	#1	Mean	4,43	5,41	8,96
		Max	65	65	89
	#2	Mean	3,95	5,20	10,88
		Max	57	66	70
Number of docks utilized	Mean	2,61	2,48	1,94	1,52
	max	5	4	3	2

Table 3 The influence of the number of forklift trucks

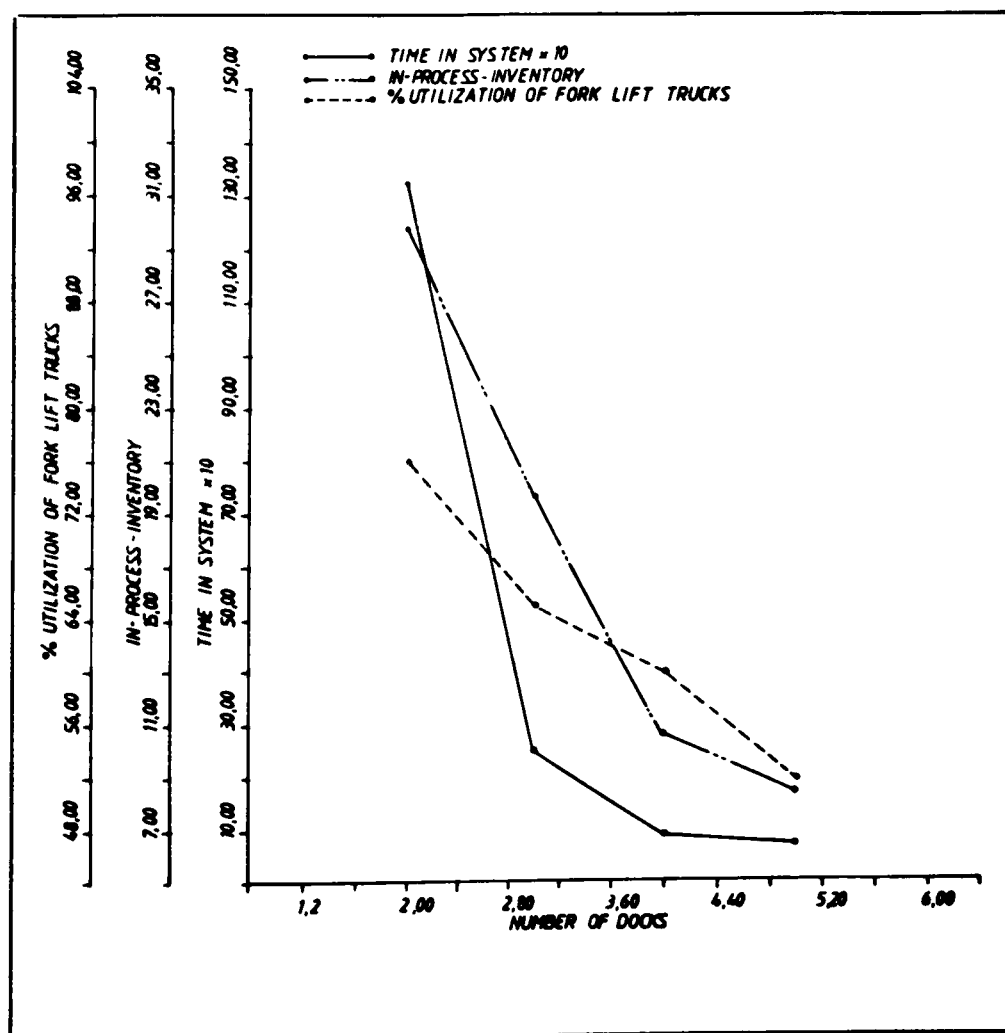
		Number of forklift trucks				
		6	5	4	3	2
Time in system	Mean	88,46	89,93	66,95	127,6	1234
	Min	55,1	51,04	69,7	80,5	150,5
	Max	209,9	189,1	134,1	257,0	3446
Number of trucks in E	Mean	2,76	2,92	1,38	4,8	75,45
	Max	7	8	4	10	140
In process inventory	#1	Mean	5,41	4,33	4,13	4,87
		Max	65	61	60	94
	#2	Mean	5,2	8,56	7,51	4,22
		Max	66	76	69	75
# of forklift trucks utilized	Mean	2,12	2,12	2,11	2,12	1,96
	Max	6	5	4	3	2

values of the number of fork-lift trucks available, and for a fixed number of four docks, the results shown in Table 3 may be obtained.

Figures 6 and 7 show some of these results in a graphical format.

Recommendations

Simulation modelling is by its very nature not a global optimization technique but is capable of evaluating the expected consequences of a particular proposed policy or decision and may therefore be seen as a decision

**Figure 6**

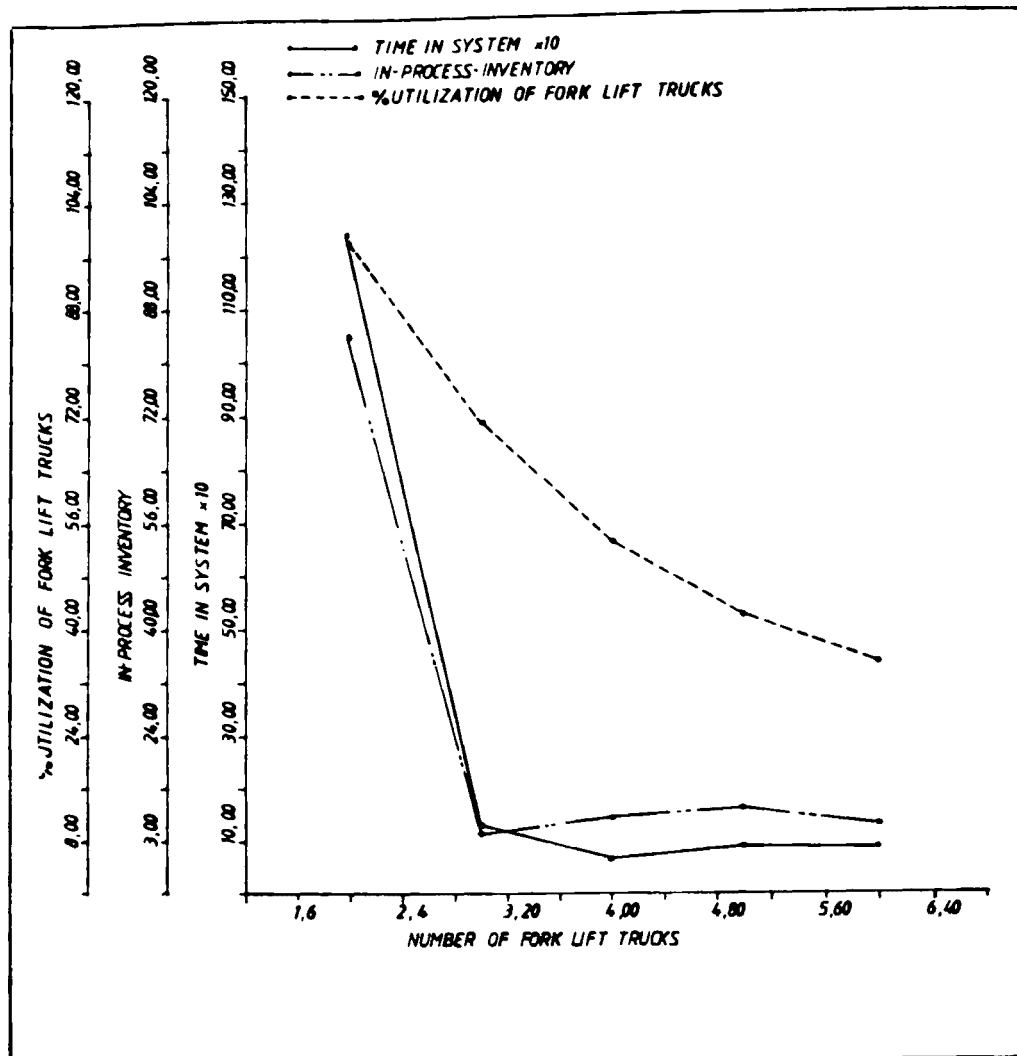


Figure 7

support tool. The results of experiments with a simulation model, such as those shown in Tables 1 and 2, may be used as input to a final optimization and decision-making process. This process should include consideration of factors not included in the model such as the construction cost of additional loading/unloading docks, the capital and operating cost of forklift trucks and the cost of in-process-inventory. Less easily quantifiable factors, such as company policy with regard to future plant expansion and market share, may also be taken into account. In this way the results from the simulation model may be used as a baseline for final recommendations and decision making.

Conclusion

The simulation technique, and in particular an advanced simulation language like SLAM II, is, when used intelligently, a very powerful system modelling approach capable of providing an efficient systems analysis tool.

Only a limited part of the available SLAM II modelling capabilities were used to build a valid and useful model of a relatively complex materials handling system. The model may be used to provide the necessary information to support the correct decisions regarding system design and operation. It should however be

realized that this simulation model, as is the case with all modelling approaches, is only an abstraction of the real system as perceived by the analyst. It may therefore be appropriate to introduce a number of enhancements and/or extensions to the model to produce a more detailed and hopefully more correct model.

The capability of the simulation modelling approach whereby the level of detail of a model may be easily and in a heuristic way adapted over a wide range, may be, in our opinion, one of the most significant advantages of the simulation approach in relation to other well known modelling approaches.

References

- Catalog of simulation software. 1983. *Simulation*, p.156.
- Gordon, G. 1975. *The application of GPSS V to discrete system simulation*. Prentice-Hall Inc.
- Pegden, C.D. & Pritsker, A.A.B. 1979. SLAM : Simulation language for alternative modeling. *Simulation*, November, p.145.
- Pritsker, A.A.B. & Pegden, C.D. 1979. *Introduction to Simulation and SLAM*. John Wiley and Sons.
- Pritsker, A.A.B. 1982. Applications of SLAM. *IIE Transactions*, March, p.70.