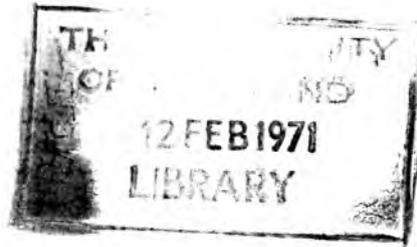


Z 658.072  
N 55



# NEWSLETTER

*Operational Research Society of New Zealand (Inc.)*



AUGUST 1970

VOL. 5. NO. 4.

THE HOCUS SIMULATION LANGUAGE  
AND SOME APPLICATIONS

H.B. MOORE

Paper presented at the 1970 ORSOC Conference.

(PERSONAL NOTES)

AIM

HOCUS stands for Hand or Computer Universal Simulator. The aim of this paper is to describe the concepts and some of the uses and limitations of the HOCUS simulation system.

SUMMARY

HOCUS is a general purpose Fortran based simulation language. It uses three concepts; that of an entity, an activity, and a queue. Entities can be alternating queues and activities, and most systems can be modelled with suitable interactions between such life-cycles. Once a model has been built up, the simulation programme can easily be set up by entering the flow-chart details directly as input. Output is time oriented and fairly comprehensive, and may be extracted as required. Rerun features are available.

Due to the nature of the language HOCUS is best suited for smaller simulations, although large systems can be effectively simulated provided the commonest time unit involved is not in terms of seconds.

In general, HOCUS appears to be a fairly versatile, easily understood simulation language that is very easy to use. It is not particularly suitable for simulating large or material-oriented systems, but for the analyst wishing to solve a range of alternative machine allocation, vehicle movement or plant location problems for example, it should prove time-saving and effective.

INTRODUCTION

Simulation is used by the analyst when he wishes to determine the effects of possible changes in an existing system, when he cannot calculate these effects directly.

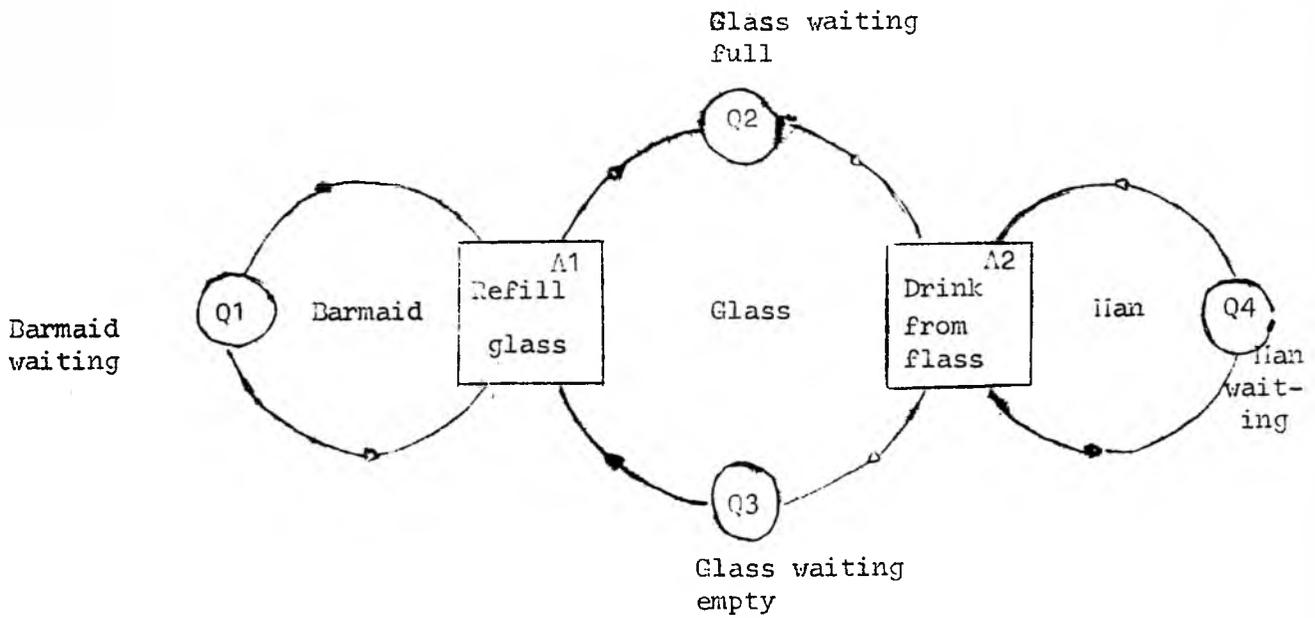
Most simulations conducted for this purpose involve the use of a computer programme. This is either user-written or supplied by consultants or computer manufacturers. User-written programmes often suffer from debugging problems, whereas commonly available simulation languages are often difficult to become familiar with, and may not be very general in their approach. The authors of HOCUS have tried to make their system as simple as possible in the hope of attracting a wider range of users. The programme was originally written in Fortran for the ICI 1130, but versions are now available for most computers with Fortran compilers.

GENERAL CONCEPTS

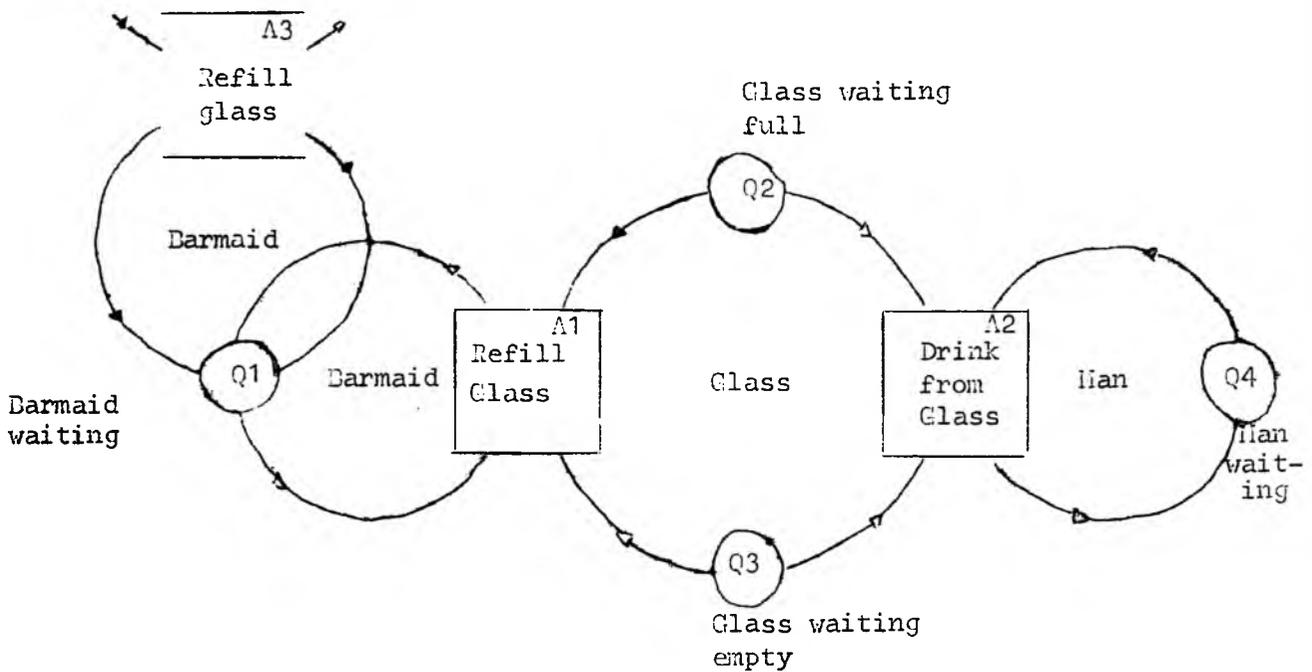
The concepts used as a basis for the HOCUS language are the entity, the activity, and the queue. An 'entity' is a unit of material in the system, e.g. a man, truck, crane or log. An 'activity' is a time consuming process that involves one or more such entities for a given period, e.g. a crane loading a truck, or a log being sawn. A queue is the idle state into which an entity passes once it has completed an activity.

Any system can be modelled using a suitable combination of entities, activities, and queues. Each entity can be considered as having a life-cycle consisting of alternating queues and activities. Each activity takes one or a number of entities from one set of queues; processes them and passes them onto another set of queues.

From here they may be taken by yet another activity, and so on. For example, consider the simple model of a man, at a bar, drinking beer. There are three entities: the man, the beer glass and the barmaid who refills it. There are two activities: the man drinking and the barmaid refilling the glass. There are four queues: the glass waiting to be filled, the full glass waiting to be drunk, the man waiting for his glass to be refilled and the barmaid waiting to refill it. We may present this as follows:



Notice that there are three life cycles involved, each containing equal numbers of queues and activities. The system can be extended to the case where the barmaid has more than one customer :



Thus the entity "barmaid" can go from one queue to any of several activities, which raises the problem of activity priority, Priorities must be ascribed when the system is formulated.

Simulation time may advance in either of two ways. Firstly by moving the clock ahead with uniform time increments and scanning the system for possible changes at each step. Secondly by scanning the finishing times of the activities in the system and jumping the clock to the earliest of these times; at that stage scanning for possible changes.

JOCUS uses this alternative method, which is obviously faster if the activities endure for more than one time unit.

ADVANCE OF SIMULATION

Once the earliest finishing time has been determined and the clock jumped to that time, there are three steps in the advance of the simulation.

- 1) End the activity or activities involved and send their entities on to their respective destination queues.
- 2) See whether any new activities can start, and if possible start them.
- 3) From the new set of activity finishing times, select the next earliest.

SIMULATION CONCEPTS IN DETAIL

Entity is the working part of the simulation model. It may be a real object, such as a man or machine, or it may be a conceptual device. An entity name may apply to a single entity or to a class of entities if they are indistinguishable, e.g., the entity name "log" may refer to each member of a group of logs. An entity is defined as such by writing its name in an entity table together with the number of like entities. The name must not be more than 8 characters long.

Associated with each entity type there are two clocks which can be used to record the time the entity takes in any part of its life cycle. This data can be used directly as output, or to influence the duration of other activities.

Queues. Each queue must have a name not more than 8 characters long, but distinct from any entity name. Queues must be numbered consecutively from 1 to n. It is by these numbers that source and destination queues are identified in an activity. It is possible to restrict the number of entities that can exist in any queue. If the queue is full as it may block any activity which would normally feed it.

Activities. Each activity must have a distinctive name not more than 12 characters in length. The activities will be assigned a priority according to their activity number, the lowest number having the highest priority. The computer will assign priority numbers in the order in which it reads the activities.

There are three ways in which the duration of an activity can be assigned, by giving it :-

- 1) A constant time component. This means that, if no other conditions are applied, the activity will always take this time.
- 2) A distribution or timetable component. The activity duration may be sampled from a timetable or cumulative time distribution. Any mathematical or empirical distribution can be inserted into the simulation by this means.
- 3) A coefficient time component. This is a specified percentage of the time last recorded by either of the clocks associated with a given entity, and can be used to either lengthen or shorten the duration of an activity. Any of these components may be used in conjunction with the others. An activity duration must not be allowed to become negative.

1) The HOCUS Flowchart

Using combinations of entity life-cycles, a physical system can be modelled in a HOCUS flowchart in which details of activity conditions and queue sizes may be entered. This flowchart can be used as a basis for a hand simulation to aid debugging. Once it is complete, entries on the flowchart may be coded directly onto forms to be punched as input for the computer programme.

2) Input to the programme involves 6 steps.

- 1) Specify the entity types and numbers of entities in each type.
- 2) Define the queues in the order in which they are to be numbered.
- 3) Define any time distribution or timetable used by the activities.
- 4) Define the activities themselves.
- 5) Initial conditions - all entities used in the system must be accounted for. These are placed in appropriate starting queues.
- 6) Once the programme has been loaded into the computer, any histograms required must be specified when the programme asks for them.

3. Output features include

- 1) A listing of activities ended as time progresses.
- 2) A listing of activities in progress.
- 3) A listing of activities starting as time progresses.
- 4) Numbers of entities in each queue.
- 5) Contents of each queue.
- 6) An entity trace.
- 7) An activity run count report.
- 8) A map contents report - this will indicate queue sizes and activity states in a layout corresponding to the HOCUS flowchart.
- 9) Histograms, if requested, of the times taken by entity types in any previously defined section of the system. These times are those recorded by the A and B clocks assigned to each entity type.

## 4) Rerun features with a new or previously used series of random numbers are available.

APPLICATIONS

H.Z. Forest Products Ltd., has had the HOCUS system since early 1970. To date it has been used by the Operations Research, Work Study, Forestry, and New Projects departments. Applications so far include :

- 1) An investigation into the economics of providing a second tractor in certain forestry logging operations. Briefly, the problem was that with a single tractor, the men felling trees, attaching the logs to the tractor, processing the logs on the dump and loading the trucks, were under-utilised. The HOCUS simulation showed that 2 tractors could be operated with negligible interference and also indicated the number of additional men required at the dump and felling areas. The simulation indicated productivity gains could be made by using 2 tractors, and actual trials to date are following the HOCUS results quite closely.
- 2) A study of one of the pulp log handling systems, to see whether an increased intermediate storage will significantly relieve peak loads on part of the chipping system.

- 3) A study to determine whether the present layout of the sawmill log end is of sufficient capacity to handle the extra number of logs required for the next mill expansion. A new distribution of log sizes is to be supplied to the mill in future and specifically the problem involves determining the loading on the log chipping system once the new log distribution comes into effect.
- 4) An investigation into light vehicle usage on and around Kinleith site to see whether the numbers of light vehicles operated by the company could be optimised.

#### LIMITATIONS OF HOCUS

- 1) HOCUS is a time-oriented simulation language, i.e. it is easiest to use when simulating vehicles, machinery, or personnel for example. Its design makes it difficult to simulate material-oriented systems such as storage capacity problems, e.g. log storage (discrete) or pulp storage (continuous).  
  
As a consequence output is time-oriented and the only summarised output available is the activity run count and the histograms of the entity times. For information on entity distributions in queues and queue size statistics, one must have the computer print and output continuously, and summarise it manually at the end of the run. This considerably retards the advance of simulation time. Most frequently, however, time statistics are the only ones of importance.
- 2) In any system of reasonable size HOCUS may prove too small. Certain features can be extended at the expense of others, but this in itself may not be enough. The analyst can re-arrange the model to fit the package, but the increased sophistication of programme necessary may obscure the important simplicity of the flowchart. As a result one of the major advantages of HOCUS - that of being easily understood - is lost.
- 3) HOCUS is not particularly suitable for modelling fast systems for a long time. If entities move in seconds rather than minutes or hours, the analyst may find the simulation run taking longer than real time. This of course is very dependent on the complexity of the system being modelled.

#### ADVANTAGES

The advantages of HOCUS over a Fortran Simulation are most evident when a small system is being modelled (say, less than 40 activities). Both simulations require the same data but the HOCUS system appears in general to be easier to set up and verify than its Fortran counterpart. Once the model is complete it can be worked through manually (hence Hand or Computer Universal Simulator) to aid debugging, thus saving computer time.

No knowledge of Fortran is necessary in order to simulate with HOCUS and because of the conversational mode of instructions in the programme, it is not necessary to be greatly familiar with the computer either. A very comprehensive edit programme prints out diagnostic messages at the start of a run.

#### REFERENCES

The HOCUS system is supplied by the PE Consulting Group Limited, Management Sciences Division, Wick Road, Egham, Surrey, England. It was designed by Mr Robin Hills of Imperial College in association with the PE Consulting Group.

An article describing the system appears in both Data Processing, May-June 1968, and Science Journal, February 1969.

PERSONAL NOTES.

PROF. G.A. VIGNAUX

The President of the Operational Research Society, Tony Vignaux, will be spending 3½ months from the middle of November, on Refresher Leave from V.U.W. He intends to travel around the U.S.A. and the U.K. investigating O.R. courses provided and observing different O.R. groups at universities, research establishments and within industries in action. He will also be taking a keen interest in computing and computer science.

DR. H. BARR

Hugh Barr of Applied Mathematics Division, D.S.I.R., will be visiting North America from September to November to investigate current research on mathematical models of transport systems. He hopes to study multi-mode allocation models and applications of O.R. to transport problems.

He will spend most of his time in Washington with the Department of Transportation and World Bank. Amongst others, he will also be visiting the California Department of Transportation in San Francisco, North Western University in Chicago, the Canadian Transport Commission in Ottawa and the Port of New York Authority.

MR J.A. JORDAN

John Jordan, Staff Manager of ICL (NZ) Ltd., will be travelling to the U.K. in October to attend a Training Conference of domestic and overseas officers of ICL companies whose functions include responsibility for staff and customer training programmes. The purpose of the conference is to review developments in training techniques and the processes whereby ICL trains its staff to centralised standards on a world-wide basis.

On his way to the U.K. he will be visiting North American Rockwell Corporation in Los Angeles and MIT in Boston to discuss management training. He may also visit the Harvard Computation Laboratory. Returning from the U.K. he will go to Moscow for one day, then Tokyo, Osaka for a day at Expo, Hong Kong and Sydney.

As John is chairman of the N.Z. O.R. Society's Education Committee arrangements are being made for contact with Mrs Kinnaid, Secretary of the U.K. O.R. Society in London, to examine the activities of the Education Committee of the U.K. Society.