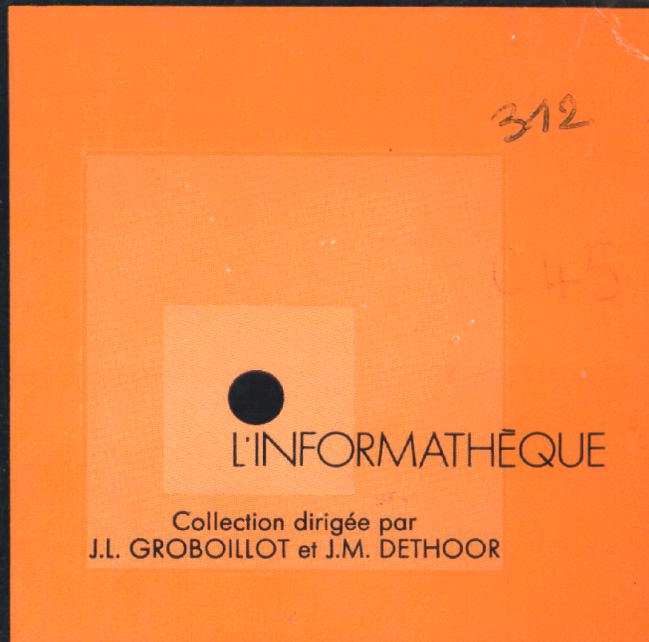


J.-R. SULZER,
P. BOUTEILLE, P. ARQUIÈ,
J.-L. BOUCHET, P.-Y. DURAND



la simulation

initiation pratique au GPSS



ENTREPRISE MODERNE D'ÉDITION

La simulation, initiation pratique au G P S S

(General Purpose Simulation System)

par

J. R. SULZER, H.E.C.

P. BOUTEILLE, H.E.C.

P. ARQUIE, H.E.C.

J. L. BOUCHET, H.E.C.

P. Y. DURAND, H.E.C.

Préface de A. KAUFMANN

ENTREPRISE MODERNE D'ÉDITION, 4, rue Cambon, Paris-1^{er}

Sommaire

Avant-propos de J. Marly	IX
Préface de A. Kaufmann	XI
Introduction : La simulation et le GPSS	XIII
1 — Les outils mathématiques du GPSS	1
I. Paramètres — SNA — Adressage indirect	1
II. Utilisations des noms symboliques	5
III. Expressions-variables	8
IV. Fonctions GPSS	15
2 — Création et destruction des transactions	27
I. Bloc GENERATE	27
II. Bloc PRIORITY	33
III. Bloc TERMINATE	35
IV. Bloc ADVANCE	38
V. Blocs SPLIT — ASSEMBLE — GATHER	43
3 — Unités de traitement et ruptures de séquence	51
I. Les stations simples	51
II. Bloc TRANSFER	60
III. Bloc LOOP	69
IV. Bloc MATCH	71
V. Files d'attente	76
VI. Stations multiples	90

4 — Contrôle logique du flux des transactions	97
I. Variables booléennes	97
II. Bloc ASSIGN	104
III. Bloc TEST	112
IV. Blocs GATE	119
V. Interrupteurs logiques	126
VI. Groupes en GPSS	133
5 — Le contrôle de l'exploitation et de l'output	143
I. Bloc MARK	143
II. Tables — Bloc TABULATE.	144
III. Bloc SAVEVALUE	152
IV. Cartes de contrôle GPSS	156
Exemple final	166
Conclusion	173
Annexes	
1 — Tableau récapitulatif des SNA	175
2 — Lexique	178
3 — Cartes de contrôle O.S.	179
4 — Liste des erreurs GPSS.	180

Avant-propos

Dirigée par J.L. GROBOILLOT et J.M. DETHOOR, professeurs du Centre de Calcul que la Chambre de Commerce et d'Industrie a installé voici deux ans, sur son domaine de Jouy-en-Josas, cette collection a pour objectif de fournir à tous, étudiants, cadres d'entreprises, fonctionnaires, etc. le moyen de s'initier par étapes à l'outil informatique, aux techniques et au langage que son maniement implique et aux diverses utilisations qu'il permet.

De conception très pédagogique, cette collection se caractérise par un enseignement progressif, l'abondance des exemples de réalisation sur ordinateur, la qualité des exercices et des corrigés. Chaque ouvrage offre ainsi à ses lecteurs une initiation, une expérience pratique et la possibilité de maîtriser le sujet traité.

Plusieurs volumes ont pour auteurs de jeunes garçons fraîchement diplômés de l'École des Hautes Études Commerciales, et c'est un agréable privilège au directeur de cette École que de souhaiter à ces travaux tout le succès qu'ils méritent.

Les jeunes, en effet, sont tout naturellement attirés par cette nouvelle science, elle-même très jeune, mais dont les potentialités sont à la mesure de leurs ambitions légitimes et dont le rôle dans les sociétés économiques apparaît comme devant être décisif. Ayant bénéficié des conditions particulièrement favorables que leur faisaient l'École et la Chambre de Commerce et d'Industrie de Paris, de l'existence d'une équipe de professeurs dynamiques et de la présence d'un ordinateur sur place, il est normal que ces élèves diffusent largement les connaissances fondamentales dont ils se sont rendus maîtres. Ils aident ainsi les utilisateurs à acquérir les connaissances et les habitudes mentales qui constitueront nécessairement demain le langage commun à tous ceux qui se trouveront lancés dans l'action.

C'est pour nous tous le moyen aussi de contribuer, même modestement, au développement des techniques de gestion et à la préparation des responsables de demain.

Jean MARTY

Directeur de
l'École des Hautes Études Commerciales

Préface

L'idée de simuler ou, si l'on préfère, de réaliser des expériences par simulacre, est aussi ancienne que l'homme et a été une des premières manifestations de l'intelligence de notre espèce. Les sumériens faisaient des modèles de leurs canaux d'irrigation, les anciens égyptiens faisaient des modèles de leurs projets de pyramides, l'idée de maquette et de modèle réduit peut être retrouvée dans tous les travaux des ingénieurs civils et militaires le long de toute l'histoire connue. Une pièce de théâtre est une simulation d'une suite de situations, un match de football ou de rugby est une simulation de bataille selon des règles très particulières, toutes ces simulations sont éducatives et stimulent l'intelligence face aux situations complexes du réel. De telles simulations sont donc réalisées d'une façon artificielle par rapport au réel observé et analysé ; elles peuvent se rapporter à des phénomènes physiques, des phénomènes sociaux, économiques, ou même cosmiques.

Avec l'ordinateur, le concept de simulation a pris un aspect nouveau qui en multiplie les applications et la facilité. Les phénomènes séquentiels et temporels, c'est-à-dire s'écoulant dans le continuum temps, peuvent être répétés, après une simplification convenable, à partir d'un programme introduit dans l'ordinateur. Le phénomène à étudier peut être répété à partir des instructions fournies mais à des vitesses beaucoup plus grandes que dans la réalité et la nature de ce phénomène. Ce qui se passe en un jour peut être contracté à quelques milli-secondes, cette contraction se situe donc à plusieurs millions, ce qui est énorme. Ces simulations, on pourrait d'ailleurs les faire à la main, par des hommes qui joueraient le rôle d'éléments logiques d'ordinateurs, dans une sorte de technodrame, mais le nombre d'hommes à utiliser et le temps nécessaire seraient prohibitifs. Tout ce qu'un ordinateur fait, un groupe d'hommes peut le faire, même les simulations les plus compliquées, mais la lenteur de cette « machine humaine » la rend impropre. Certains diront, pourquoi ne pas faire les expériences sur le réel et le vécu ? On s'y refuse parce que les phénomènes économiques et sociaux impliquent que les expériences soient faites par les hommes de ce réel, avec les indiscutables dangers psychologiques. Imaginez qu'une certaine ville soit soumise à des expériences sur la circulation des automobiles, expériences changeant chaque jour, donc chaque jour de nouvelles règles, ceci pour examiner quelles seraient les meilleures. Les habitants de cette cité ne deviendraient-ils pas fous rapidement s'ils ne se révoltaient plus rapidement encore ?

L'ordinateur permet donc de faire des expériences artificielles sur toutes sortes de phénomènes économiques et sociaux. Files d'attente, gestion de stocks, investissements, processus de distribution, de production, comportement, marketing, etc., toutes les situations correspondantes sont susceptibles d'être simulées. La taille des phénomènes à simuler ne fait plus reculer

BIBLIOTHEQUE DU CERIST

les utilisateurs d'ordinateurs : un port de commerce, un réseau urbain, une usine, un réseau interconnecté de liaisons de communications, un aéroport, sont couramment soumis à des simulations. Elles apportent des informations précieuses sur les performances des systèmes, elles permettent de comparer des politiques, voire des stratégies, elles donnent donc des possibilités d'expérimentations presque illimitées dans de nombreux domaines des sciences humaines. Rappelons toutefois que le but d'une simulation n'est généralement pas l'optimisation mais l'expérience et l'adaptation. L'optimisation est une notion à associer à l'analyse mathématique, c'est un domaine propre à la recherche opérationnelle. Rappelons encore que la simulation ne s'oppose pas à l'analyse mathématique et qu'il est toujours recommandé, avant de faire une simulation, d'épuiser les possibilités de l'analyse.

Pour effectuer une simulation sur un ordinateur, il faut le programmer. Il fut un temps où cela se faisait en langage machine et demandait un temps prohibitif. Puis on a utilisé des langages symboliques généraux, cela était bien plus facile. Enfin, on en est venu aux langages spécialisés. Parmi ceux-ci, le G.P.S.S. est un des plus efficaces, bien adapté aux processus séquentiels aléatoires, à la portée de toute personne ayant déjà programmé en langage symbolique. Il évite les instructions les plus délicates rencontrées dans les programmes symboliques construits à partir des organigrammes des processus séquentiels. Bien entendu, de tels langages spécialisés ne peuvent être absolument universels mais ils le sont suffisamment pour les classes les plus fréquemment rencontrées.

Le grand mérite de MM. SULZER, BOUTEILLE, ARQUIE, BOUCHET et DURAND, élèves de H.E.C., est d'avoir su présenter ce nouveau langage de simulation, d'une façon claire, extrêmement didactique, d'avoir présenté un grand nombre d'exemples et d'illustrations, d'avoir su informer le lecteur des principales difficultés existant dans cette programmation. Quel meilleur exemple pourrait être donné à d'autres élèves des grandes Écoles françaises : commencer à enseigner quand on est encore élève. Cette conception n'est pas paradoxale, elle est dans la nature même de la civilisation de demain. Celui qui reçoit de l'éducation a le devoir de la transmettre à son tour. Et il n'est jamais trop tôt pour le faire. Le devoir d'enseigner a pour corollaire la volonté de bien apprendre et de créer. Nos jeunes et brillants amis en ont encore apporté la preuve.

A. KAUFMANN

Professeur à l'Institut Polytechnique de Grenoble,
Conseiller scientifique de la Compagnie de
Machines Bull-General Electric.

Introduction : Le GPSS

(General Purpose Simulation System)

Simuler le réel, c'est en concevoir une représentation sur laquelle essais et mesures sont facilement réalisables.

Lorsqu'un entrepreneur conçoit une nouvelle installation ou un nouvel équipement, il peut se heurter à des difficultés plus ou moins grandes pour les essayer avant de les utiliser ou de les lancer sur le marché; dans le cas d'une automobile, on peut réaliser un prototype, que l'on teste sur route; dans le cas d'un navire, on en réalise une maquette que l'on essaie en bassin pour éprouver son hydrodynamisme. Par contre, une telle méthode d'expérimentation devient quasiment impossible lorsqu'il s'agit d'équipements d'un type unique et bien particulier, tels que supermarchés ou station-services.

Ce que par exemple une Société de Distribution de Produits Alimentaires (SDPA) peut connaître, c'est le temps que passent les clients à faire leurs achats devant tel type de rayon, le débit de portes d'entrée d'un modèle précis ou celui des guichets de paiement qu'elle emploie habituellement. Supposons qu'elle envisage de construire un nouveau supermarché qui utilise ces éléments déjà connus, mais qui soit beaucoup plus vaste et inaugure une nouvelle disposition; il sera alors très difficile à la Société de prévoir si cette nouvelle installation sera adaptée à une circulation optimale des clients. La simulation — et GPSS en particulier — permet de réaliser une telle expérimentation, comportant les trois phases suivantes :

- Il importe que l'entrepreneur ait à sa disposition des chiffres et des statistiques précis concernant les divers éléments de base qui composent l'installation à tester. Ce n'est qu'à cette condition qu'il obtiendra des résultats satisfaisants;
- Il doit ensuite réaliser un programme qui représente en quelque sorte le plan, la structure de la nouvelle installation, synthèse des éléments de base. Le programme constitue une véritable maquette, un prototype, de cette installation. Il est le noyau de l'expérimentation fictive réalisée par l'ordinateur;

— Lorsque GPSS a fini de simuler le fonctionnement de l'installation, — en un temps extrêmement court —, il délivre toutes les statistiques désirées qui s'y rapportent.

C'est ainsi qu'en modifiant certaines données du programme, en fonction de ce que permet la réalité, la SDPA pourra essayer différentes dispositions de son supermarché, et voir celle qui répond à une utilisation optimale. L'ordinateur lui permet donc de réaliser avec souplesse et facilité ce qu'elle n'aurait pas pu faire dans la réalité.

Deux exemples illustreront cette idée :

Premier exemple : Les Ponts et Chaussées vont construire une autoroute à un seul poste de péage. Leur expérience des autoroutes déjà réalisées et les données relatives au trafic routier de la région de construction leur fournissent les éléments de base suivants :

1. Il arrive une voiture toutes les 15 secondes;
2. La voiture fait la queue si le poste de péage est occupé;
3. Quand il est libre, la voiture y entre...;
4. ... et quitte donc la queue;
5. La voiture reste 30 secondes dans le poste de péage, le temps que l'automobiliste paie et que la barrière se lève;
6. La voiture libère le poste de péage;
7. La voiture disparaît au loin.

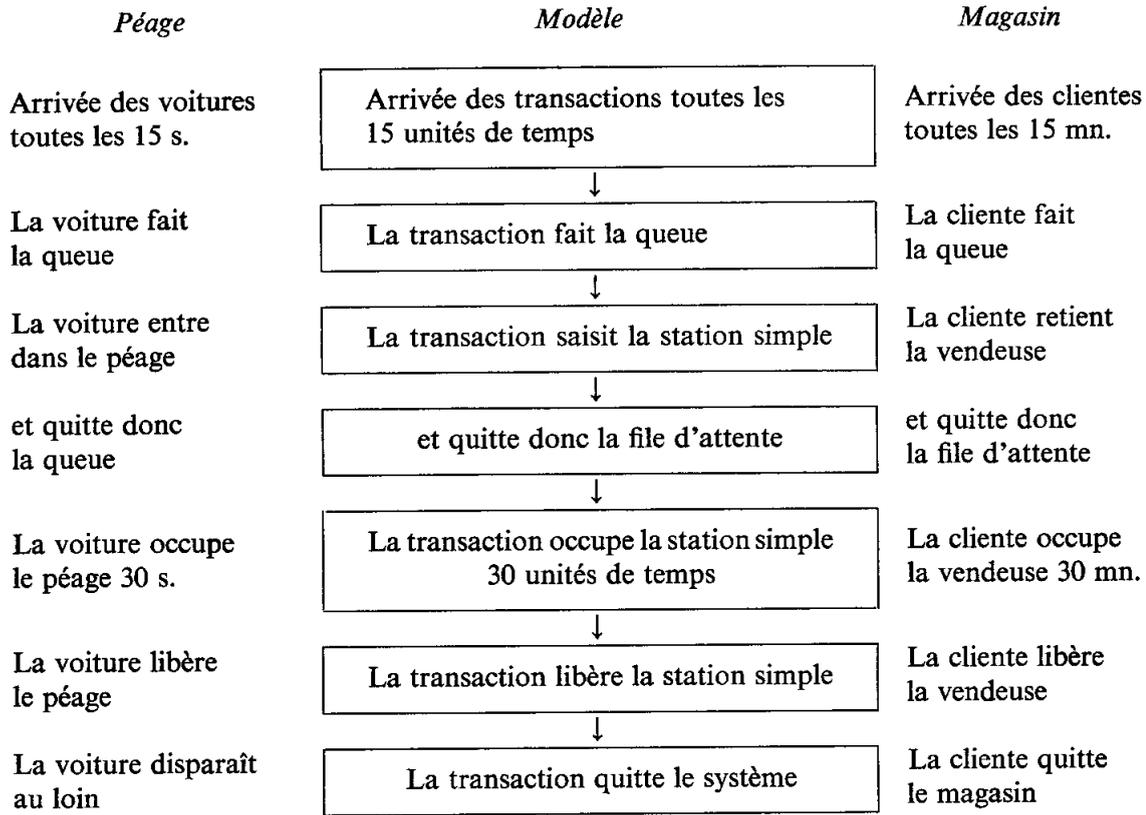
Deuxième exemple : M^{me} BACHON vient d'acheter un magasin de fourrures. La précédente propriétaire la met au courant :

1. Il arrive une cliente toutes les 15 minutes;
2. La cliente fait la queue si la vendeuse n'est pas libre;
3. Quand la vendeuse est libre, elle se met à servir la cliente suivante;
4. Qui quitte donc la file d'attente;
5. Vu le prix des fourrures, le choix est long : il faut 30 minutes pour servir une cliente;
6. La cliente libère la vendeuse;
7. La cliente quitte le magasin.

Ces deux exemples ont un certain nombre de points communs :

- Dans les deux cas, on trouve des éléments qui arrivent, se déplacent dans le système et repartent : les voitures et les clientes. Ces entités sont appelées «transactions» en GPSS;
- Dans les deux cas, ces transactions peuvent être bloquées dans une file d'attente. L'analyse de ces files d'attente permet de déceler d'éventuels goulots d'étranglement;
- Dans les deux cas, on trouve un élément qui est saisi par les transactions les unes après les autres et qui n'en accepte qu'une à la fois : le péage, la vendeuse : c'est ce qu'on appelle «station simple» en GPSS.

De plus, ces deux exemples se ramènent au même modèle de simulation :



Ces deux exemples donnent une idée des applications très nombreuses du GPSS : chaînes de production, autoroutes, ports, aérodromes, restaurants, magasins, cinémas...

Programmation d'un modèle simple

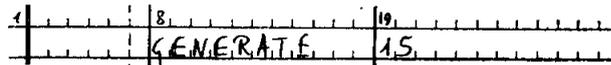
On a vu dans l'introduction que les deux problèmes posés se ramenaient à un seul modèle.

On remarque que l'action se déroule en différentes étapes. Chacune de ces étapes fera l'objet d'une instruction dans notre programme et sera représentée par un *bloc*.

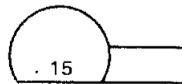
Reprenons les différentes étapes de notre modèle :

1. Il faut un bloc qui crée les transactions (voitures, clientes) pour pouvoir procéder à notre simulation. En GPSS, ce bloc est appelé GENERATE.

Pour indiquer au système qu'on utilise ce bloc, on se sert d'une carte perforée de la forme



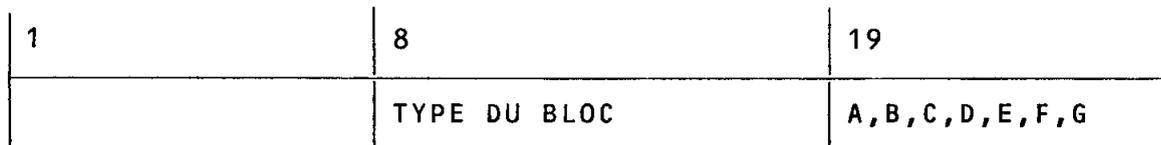
Dans un organigramme, ce bloc sera représenté par le symbole



15 signifie que GPSS crée une transaction toutes les 15 *unités de temps*. C'est l'utilisateur qui détermine cette unité de temps.

Une unité de temps ayant été choisie, l'utilisateur veillera à s'y conformer pour tout le modèle (par exemple, il ne pourra pas mélanger les minutes et les secondes dans les divers blocs).

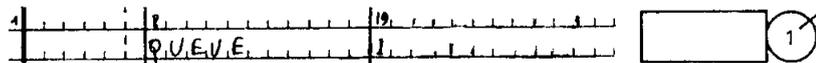
Le nombre 15 a été placé dans ce qu'on appelle *un champ* en GPSS. Pour chaque bloc, on peut utiliser jusqu'à 7 champs séparés par des virgules. Une carte perforée représentant un bloc a donc la forme suivante :



Dans le cas précédent, nous avons omis les champs B, C, D, E, F, G qui nous serviront par la suite à affiner notre simulation.

2. Les cartes qui simulent une file d'attente sont au nombre de 2 :

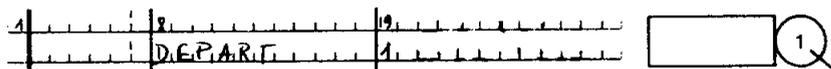
a) L'entrée dans la file d'attente au bloc QUEUE dont le but est de délivrer des statistiques relatives à la file d'attente.



Dans le modèle choisi, une seule transaction pénètre à la fois dans la file d'attente. Dans le champ A du bloc, le nombre 1 signifie qu'il s'agit de la file d'attente n° 1.

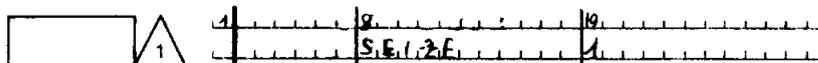
- b) Une fois que la transaction a été prise en charge par la station simple (c'est-à-dire lorsque la vendeuse ou le péage a été libéré par la transaction précédente), il faut s'assurer que la transaction quitte bien la file d'attente.

Le bloc correspondant à cette opération et le bloc DEPART



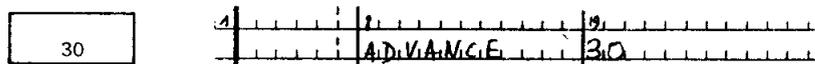
Dans le champ A, le nombre 1 signifie que la transaction quitte la file d'attente n° 1.

3. Représentons maintenant sous forme d'instructions la station simple. Tout d'abord, la transaction saisit la station-simple par l'intermédiaire du bloc SEIZE.

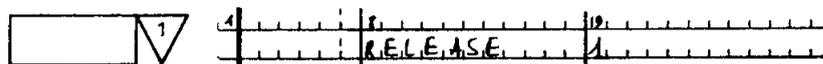


Le nombre 1 du champ A est le numéro d'identification de la station simple.

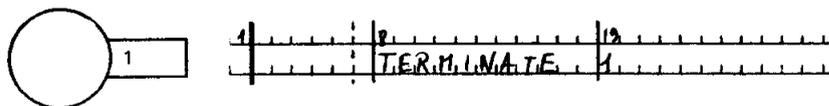
4. Puis, la transaction occupe la station simple n° 1 pendant 30 unités de temps. Le bloc ADVANCE simule cette occupation.



5. Enfin, la transaction libère la station simple n° 1 grâce au bloc RELEASE.



6. Il faut enfin montrer que la transaction quitte le système c'est-à-dire que les autos ou les clientes quittent le poste de péage ou le magasin. Le bloc TERMINATE joue ce rôle.



Chaque fois qu'une transaction pénètre dans ce bloc, elle est immédiatement détruite.

7. De plus, le nombre 1 du champ A indique que chaque fois qu'une transaction pénètre dans ce bloc, un compteur est incrémenté d'une unité. Si par exemple, on veut que 100 transactions arrivent à ce bloc pour avoir une idée précise de la solution du problème posé, on mettra la carte perforée suivante :

```
1 | 2 | 100 |
---|---|---|
| S T A R T | 1 0 0 |
```

Si on avait mis 2 dans le champ A du bloc terminate et 100 dans le champ A de la carte START, la simulation se serait arrêtée lorsque $100/2 = 50$ transactions auraient pénétré dans le bloc.

Il reste 2 cartes de contrôle à insérer dans notre programme :

- une carte pour ordonner la simulation, placée en tête du programme;

```
1 | 2 | |
---|---|---|
| S I M U L A T E | |
```

- et une carte pour préciser qu'il n'y a plus d'instructions, placée en fin de programme.

```
1 | 2 | |
---|---|---|
| E N D | |
```

Le modèle GPSS proposé dans l'introduction, se programme donc ainsi :

	1	8	19
		SIMULATE GENERATE QUEUE SEIZE DEPART ADVANCE RELEASE TERMINATE START END	15 1 1 1 1 30 1 1 100

On remarque que, dans le diagramme, le bloc SEIZE se trouve avant le bloc DEPART. La transaction doit déjà avoir saisi la station simple avant de sortir de la file d'attente, en quelque sorte par souci de sécurité.

Comment la machine va-t-elle simuler ce programme ?

Le procédé utilisé s'appelle le SCAN. Le SCAN « balaye » le programme selon une certaine chronologie. Il pourrait regarder à chaque instant t (séparé de l'instant $t + 1$ par 1 unité de temps) si toutes les transactions peuvent avancer, puis les faire avancer dans les différents blocs. Mais cela prendrait beaucoup de temps. Les inventeurs du système ont trouvé 2 méthodes pour économiser du temps :

Tout d'abord, tous les instants t ne seront pas envisagés. En effet, à une transaction on attribue ce qu'on appelle un BDT (bloc departure time ou temps de départ du bloc). Par exemple, dans le bloc GENERATE de notre exemple la transaction n° 1 a un BDT égal à 15, la seconde un BDT égal à 30.

Le système constitue ainsi une liste des BDT et ne commence qu'à l'instant où une transaction est susceptible d'avancer dans le système.

Les transactions dont le BDT est égal à l'instant t de l'horloge sont placées dans la chaîne des événements courants dite « chaîne actuelle ».

Les transactions dont le BDT est supérieur à l'instant t de l'horloge sont placées dans la chaîne des événements futurs dite « chaîne future ».

A un instant t , le SCAN ne s'occupe que des transactions placées dans la chaîne actuelle. Mais, il y en a encore trop, il distingue donc 2 catégories : celle qui sont placées dans un *statut actif* et les autres placées dans un *statut d'attente*.

Prenons un exemple. La première transaction à l'instant 15 va rentrer dans le bloc QUEUE, le bloc SEIZE puis le bloc DEPART sans rencontrer aucun obstacle. Dans l'ADVANCE BLOC son BDT va être mis à $15 + 30 = 45$.

Cette transaction est donc placée dans la chaîne future. A l'instant 30, la seconde transaction ne va pas pouvoir saisir la station simple n° 1 puisque la transaction n° 1 l'occupe. Elle restera dans la file d'attente n° 1 avec un statut d'attente; GPSS la considérera comme étant dans une *chaîne d'attente*. Ce n'est que lorsque la transaction n° 1 libérera la station simple n° 1 que la transaction n° 2 recevra automatiquement un statut actif et sera placée dans la chaîne actuelle. C'est alors que le SCAN essaiera de la faire avancer dans le système.

N.B. : Dans le champ d'opérande (colonnes 19 à 71), qui se divise en champs A, B, C, D, E, F, G, le premier blanc rencontré est considéré comme la fin de l'instruction GPSS. De ce fait il est possible de marquer des commentaires sur chaque carte d'instruction, séparés de l'instruction elle-même par au moins un blanc.

Cf. listing ci-après :

```

BLOCK          *LCC  OPERATION  A,B,C,D,E,F,G      COMMENTS          CARD
NUMBER          *      *          *          *          *          *          *          *
* CRUPE GPSS-PEC          *          *          *          *          *          *          *
* PROGRAMMATION D'UN MODELE SIMPLE
*          *          *          *          *          *          *          *
1              SIMULATE
2              GENERATE  15
3              QUEUE     1
4              SEIZE     1
5              DEPART    1
6              ADVANCE   30
7              RELEASE   1
              TERMINATE 1
              START     100
              JCB
              14

RELATIVE CLOCK          3015  ABSOLUTE CLOCK          3015
BLOCK CCLNTS
BLOCK CLURRENT          TCTAL  BLOCK CURRENT          TCTAL  BLOCK CURRENT          TOTAL  BLOCK CURRENT          TCTAL  BLOCK CURRENT          TOTAL
1          C          200
2          100          200
3          C          100
4          C          100
5          0          100
6          C          100
7          C          100

FACILITY          AVERAGE          NUMBER          AVERAGE          SEIZING          PREEMPTING
          UTILIZATION          ENTRIES          TIME/TRAN          TRANS. NC.          TRANS. NC.
1          .995          100          30.000

CLELE          MAXIMUM          AVERAGE          TCTAL          ZERO          PERCENT          AVERAGE          FAVERAGE          TABLE          CURRENT
          CONTENTS          CONTENTS          ENTRIES          ENTRIES          ZERCS          TIME/TRANS          TIME/TRANS          NUMBER          CONTENTS
1          100          49.751          200          1          .4          750.000          753.768          100
FAVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES

```

Programmation d'un modèle simple — Questions

Question 1 : Le SCAN examine-t-il l'état du système à chaque unité de temps ?

Question 2 : Dans le programme qui vous a été donné en exemple; si, dans le champ A du bloc TERMINATE on remplaçait le nombre 1 par le nombre 10; quand se terminerai la simulation ?

Question 3 : Toujours dans le programme qui vous a été donné, on se place à l'instant 64. Dans quelles chaînes sont les différentes transactions? Avec quel statut?

Question 4 : Même question, mais on change le 30 qui est dans le champ A du bloc ADVANCE, et on met 50 à la place.

Programmation d'un modèle simple — Réponses

Question 1 : Non. GPSS passe directement à l'instant où la prochaine transaction sera susceptible d'avancer dans le système.

Question 2 : La simulation se terminerait dès que 10 transactions seraient passées dans le bloc TERMINATE ($100/10 = 10$).

Question 3 :

Transaction 1 : détruite dans le bloc TERMINATE;

Transaction 2 : chaîne future (bloc ADVANCE);

Transaction 3 : } chaîne courante mais en statut d'attente, car elles sont dans la file d'attente;

Transaction 4 : }

Transaction 5 : chaîne future (temps de départ du bloc GENERATE supérieur à 64).

Question 4 :

Transaction 1 : chaîne des événements futurs (bloc ADVANCE);

Transaction 2 : chaîne des événements courants; mais en statut d'attente, car elle est dans la file d'attente, puisque la transaction 1 n'a pas libéré la station simple;

Transaction 3 : situation identique à celle de la transaction 2;

Transaction 4 : idem;

Transaction 5 : chaîne des événements futurs (temps de départ du bloc GENERATE supérieur à 64).