

C.539

SERIE A  
N° d'Ordre : 513  
N° de Série : 64

THESE  
présentée

DEVANT L'UNIVERSITE DE RENNES

U.E.R. Mathématiques et Informatique.

pour obtenir

le titre de DOCTEUR en TROISIEME CYCLE -  
Spécialité : Informatique



PAR

Robert RANNOU

Sujet de la Thèse :

Mise en oeuvre du langage Simone.  
Etude critique du langage.

Soutenue le 23 JUIN 1978 devant la Commission d'Examen.

M. J.P. VERJUS

Président

J. BEZIVIN  
O. LECARME  
J. LENFANT  
J.L. NEBUT  
A. PROFIT

}

Examineurs

## P L A N

### 0) INTRODUCTION

### 1) DESCRIPTION DU LANGAGE SIMONE

- 1.1. La notion de processus
- 1.2. La notion de moniteur
- 1.3. Synchronisation et coopération des processus - La notion de variable de condition
- 1.4. La notion de temps simulé
- 1.5. Résumé du langage

### 2) MISE EN OEUVRE DU LANGAGE SIMONE

- 2.1. Ordonnancement dans Simone - sa mise en oeuvre
  - 2.1.1. Définition de l'ordonnancement dans Simone sous forme d'un graphe d'état et de transitions entre ces états
  - 2.1.2. Ordonnancement en vrai parallélisme
    - 2.1.2.1. Mise en oeuvre sur une architecture constituée de processus identiques partageant une mémoire commune
    - 2.1.2.2. Mise en oeuvre sur machine distribuée
  - 2.1.3. Mise en oeuvre de l'ordonnancement sur un microprocesseur - Quasi-parallélisme
  - 2.1.4. Structure de la file de temps et des files de condition
- 2.2. Organisation d'un programme Pascal
- 2.3. Organisation générale d'un programme Simone
- 2.4. Allocation et récupération de l'espace des processus
  - 2.4.1. Gestion de la mémoire utilisée dans le compilateur rennais
  - 2.4.2. Introduction dans le langage de limitations sur le nombre de processus activables ou pouvant être actifs au même moment
  - 2.4.3. Eviter les restrictions sur la récursivité

- 2.5. Calcul de la taille de la zone d'activation d'un processus
    - 2.5.1. Schéma général du calcul d'espace
    - 2.5.2. Problèmes dus à la récursivité
    - 2.5.3. Problèmes annexes du calcul de l'espace
  - 2.6. Mise en œuvre des moniteurs
    - 2.6.1. Initialisation des moniteurs
    - 2.6.2. Appels des procédures et fonctions d'entrée d'un moniteur
    - 2.6.3. Terminaison d'un programme et durée de vie d'un moniteur
    - 2.6.4. Traitement syntaxique des moniteurs et de leurs procédures et fonctions
  - 2.7. Problème annexe - L'instruction "stop"
  - 2.8. Les directives d'optimisation qu'il serait possible de mettre en œuvre
    - 2.8.1. Réduction de la taille des programmes
    - 2.8.2. Amélioration de la rapidité d'exécution
- 3) PROGRAMMATION DE SIMULATION A EVENEMENTS DISCRETS
- 3.0. Exemple de travail
  - 3.1. Utilisation d'un langage évènementiel
    - 3.1.1. La notion d'évènements
    - 3.1.2. Définition d'un langage évènementiel basé sur Pascal
    - 3.1.3. Mise en œuvre du langage Pascal évènementiel
    - 3.1.4. Programmation de l'exemple de travail avec le langage Pascal évènementiel
    - 3.1.5. Extensions au langage
    - 3.1.6. Comparaison avec SIMSCRIPT
  - 3.2. Le langage Simone : son utilisation en simulation et ses limites
    - 3.2.1. Programmation de l'exemple de travail avec le langage Simone
    - 3.2.2. Le problème de la préemption
    - 3.2.3. Les extensions à Simone
- 4) ETUDE CRITIQUE DES MONITEURS ET DES VARIABLES DE CONDITION
- 4.1. Les moniteurs et les variables de condition en tant qu'outils de synchronisation

4.2. Le problème des appels entre moniteurs

4.2.1. L'approche prise avec le langage Simone

4.2.2. Pourquoi les moniteurs ?

4.2.3. Quel (ou quels) autres outils de structuration faut-il ?

5) CONCLUSION

ANNEXE 1 - Contenu du bloc de contrôle d'un processus

ANNEXE 2 - Description du noyau

ANNEXE 3 - Problèmes des 8 reines sur un échiquier

ANNEXE 4 - Liste des fonctions de générations de nombre aléatoires