

Simulation du déroulement logique de programmes de commande de spectromètres à neutrons

Ph. BLANCHARD, Ph. LEDEBT and M. TAESCHNER
Institut Laue-Langevin, Grenoble, France

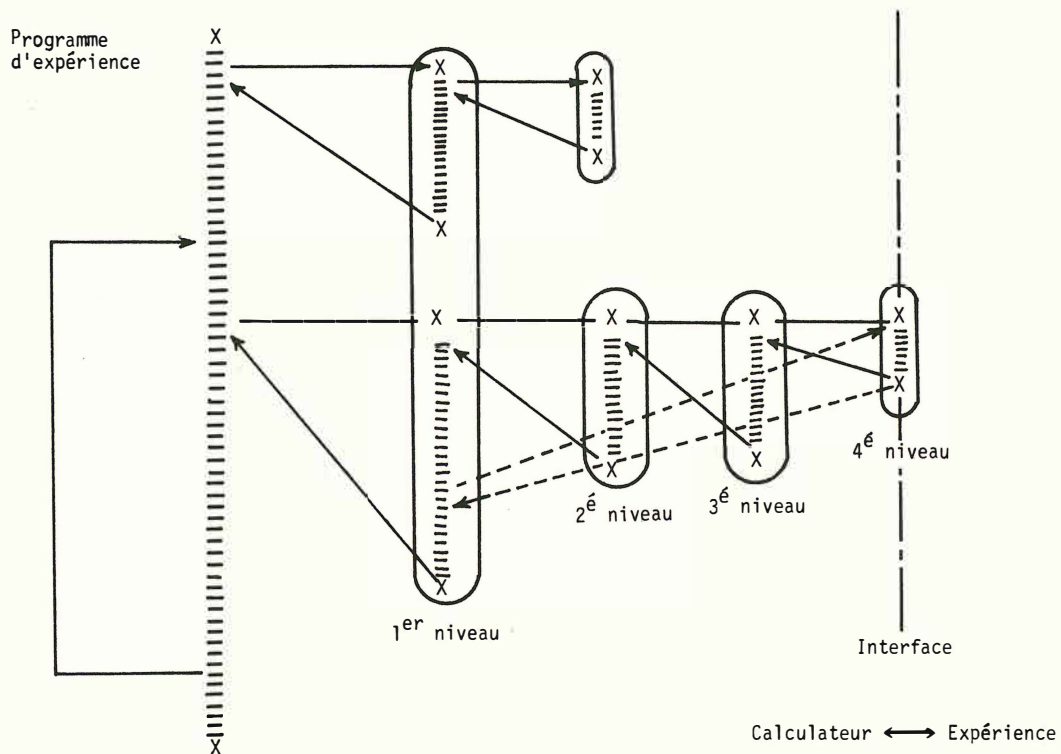
1 - Generalités - Introduction

Ce système de simulation est supporté par un terminal 2741, connecté sur le matériel IBM 360/67 de l'Institut de Mathématiques Appliquées de l'Université de Grenoble. Ce ordinateur fonctionne en time-sharing, avec le système CP/CMS, basé sur l'utilisation des machines virtuelles en mode conversationnel. Chaque utilisateur a donc à sa disposition, un ordinateur virtuel, type 360/40 avec imprimante virtuelle lecteur cartes/perfo cartes virtuel, etc.

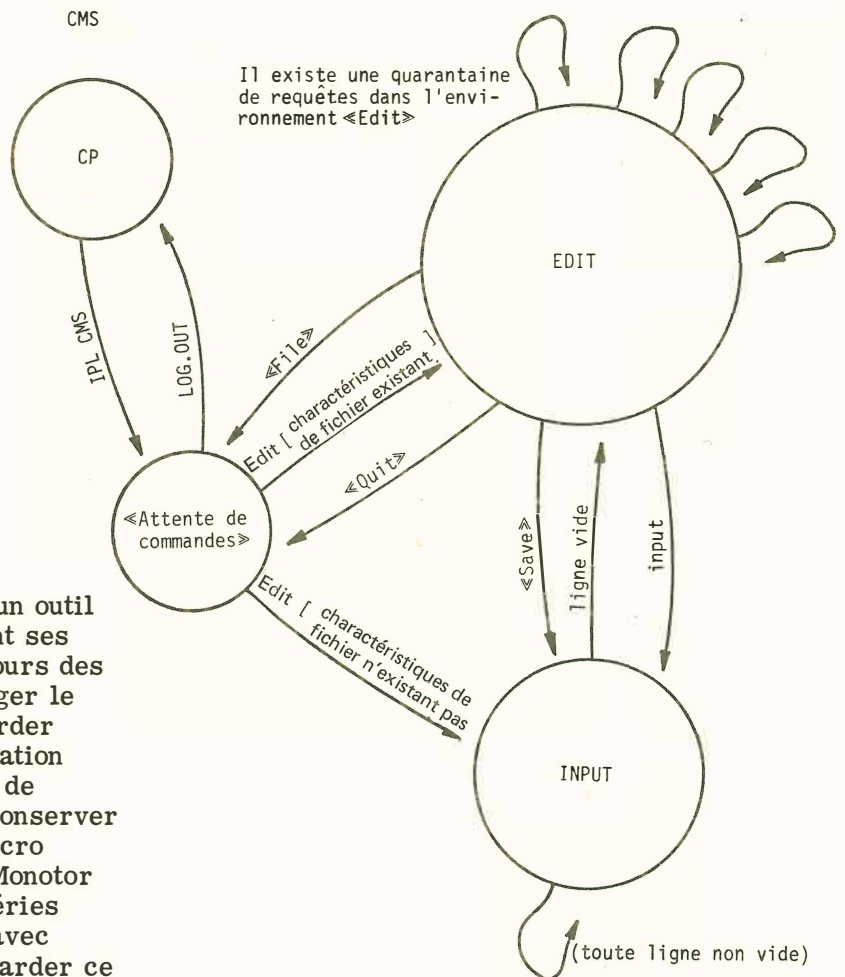
Nous allons regarder maintenant le système réel de gestion des expériences.

Description du Système Carine

Chaque ordinateur de processus gère six spectromètres par l'intermédiaire de programmes FORTRAN temps réel. Une bibliothèque de sous-programmes assembleurs, gère les entrées/sorties, et chaque sous-programme peut être appelé par le FORTRAN. Chaque utilisateur dispose auprès de son expérience, d'un télétype pour démarrer, contrôler, arrêter ses programmes et suivre ainsi l'évolution de sa manipulation. Les transferts d'information ordinateur-expérience et vice versa, sont assurés par l'intermédiaire d'un standard électronique, le CAMAC.



Diapositive 1 Diagramme général



Diapositive 2 Système CP/CMS

Simulation

Notre but était de fournir à l'utilisateur un outil lui permettant de créer et mettre au point ses programmes d'expériences, sans le secours des calculateurs de processus, afin de soulager le background de ces systèmes. Afin de garder toute son efficacité, le système de simulation garde la même structure de dialogue, et de possibilités que le système réel. Pour conserver cette structure, nous avons utilisé le macro langage de commande CMS (Cambridge Monotor System), qui permet de remplacer les séries d'ordre par une seule commande EXEC avec paramètres. Nous allons maintenant regarder ce qu'est le système CP/CMS.

11 - Système CP/CMS

CP: Le système générateur de machines virtuelles CP 67 a été conçu et développé par le Centre Scientifique IBM de Cambridge en collaboration avec le Lincoln Laboratory du M.I.T. La version initiale a été mise en service en Avril 1968. CP 67 permet de faire fonctionner sur un 360/67 un nombre quelconque de machines virtuelles qui sont des 360 standards. La mémoire centrale de cette machine virtuelle peut être plus grande que celle du 360/67 utilisé. L'activation d'une machine virtuelle se fait à partir d'une console terminale connectée au 360/67 grâce à son nom et son mot de passe. Une fois la machine activée, un système de programmation est chargé en mémoire virtuelle par utilisation du bouton (simulé) 'chargement'. A partir de ce moment, l'utilisateur dialogue directement avec le système qu'il vient de charger. Dans le cas qui nous concerne, nous chargeons un système CMS, mais d'autres systèmes sont disponibles. (APL - OS - etc.).

CMS: Cambridge Moniteur System). Le système CMS donne à l'utilisateur la possibilité d'employer, en mode conversationnel un ordinateur IBM 360,

donc une machine virtuelle générée par CP/67. L'avantage de cette solution est de mettre à disposition d'un utilisateur les vastes possibilités d'un gros ordinateur: grande mémoire, unités d'entrée sortie rapide, puissance de calcul, etc.

Immédiatement après initialisation, la machine CMS atteint un état stable particulier, appelé 'attente de commandes'. Les commandes disponibles appartiennent à deux familles:

1. Celles dont l'exécution provoque un changement d'état.
2. Celles qui après exécution impliquent un retour immédiat à l'état stable de base.

Différents états stables permettent d'entrer des fichiers, de corriger des fichiers, d'exécuter des programmes EDIT - INPUT - ATTENTE DE COMMANDES.

Chaque commande rencontrée est considérée comme phase élémentaire, d'un langage particulier et est exécutée immédiatement. L'idée de pouvoir conserver (donc de réutiliser) un groupe de commandes en lui donnant un nom collectif vient naturellement à l'esprit. On a donc un fichier de commandes et il suffit alors de frapper EXEC < nom de fichier > pour que le système aille automatiquement lire ce fichier, et en interpréter

le contenu. C'est cette possibilité que nous avons utilisée pour la simulation, afin de conserver le même dialogue que sur les calculateurs de processus. Le langage utilisé est un macro langage de commandes CMS. EXEC est donc une commande de CMS permettant l'exécution de la séquence de commandes contenue dans le fichier portant le nom de cette macro-commande.

111 - SIMUL

Le système comprend Trois parties essentielles:

1. Identification de l'utilisateur.
2. Analyses des commandes.
3. Exécution des commandes.

L'identification utilisateur comprend l'appel du système SIMUL sous CMS, puis la reconnaissance de l'utilisateur (par son nom d'expérience). Ensuite on exécute la concaténation des bibliothèques nécessaires, bibliothèques système et utilisateurs.

L'analyse des commandes, dirige vers les options Background-Foreground, (comme sur les calculateurs de processus) puis analyses les requêtes de l'utilisateur, et prépare le travail (créations de fichiers intermédiaires, appel de programmes, etc...), avant de donner la place à l'exécution.

L'exécution correspond au travail proprement dit, et peut aussi se décomposer en trois parties:

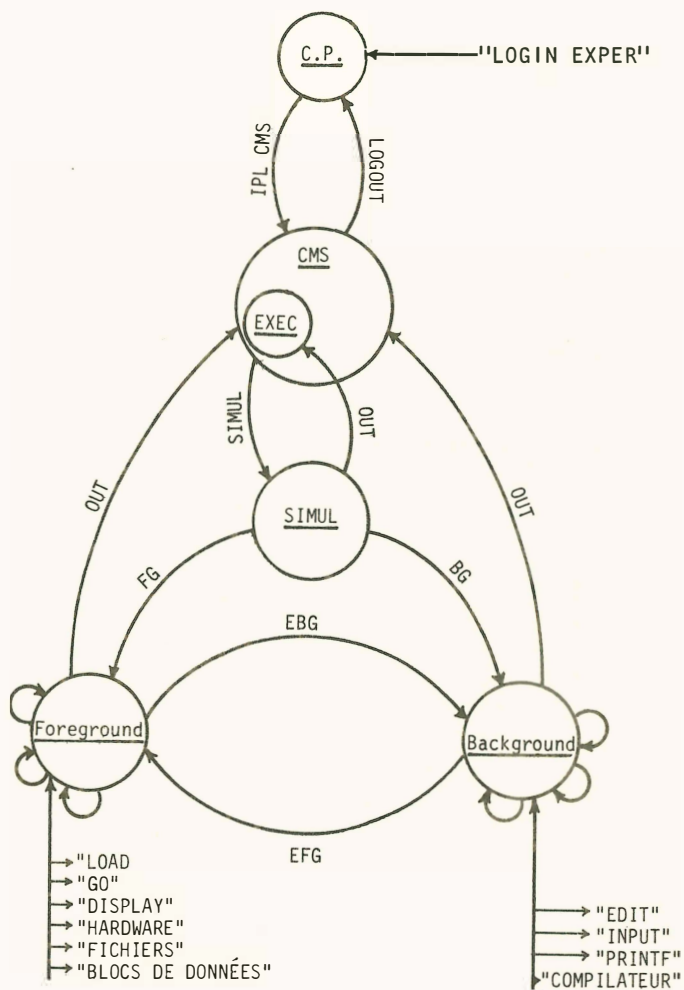
1. Exécutions d'ordres systèmes (CMS).
2. Exécutions de programmes FORTRAN système.
3. Exécutions de simulation expérience.

La partie BACKGROUND permet la création et la correction de programmes FORTRAN utilisateur, mais ne permet pas l'exécution, les modules temps réels étant seulement disponibles en exécution dans la partie FOREGROUND.

Dans l'environnement Background, les opérations suivantes sont disponibles:

1. Création d'un programme ou sous-programme FORTRAN.....(EPC)
2. Correction d'un programme ou sous-programme FORTRAN..... (CPG)
3. Liste d'un programme FORTRAN... (SPC)
4. Liste d'un sous-programme FORTRAN..... (SSC)
5. Compilation et génération d'un module..... (COP)
6. Compilation sous-programme (COS)
7. Lecture programme sur ruban (EPR)
8. Lecture sous programme sur ruban. (SSR)
9. Transfert dans l'environnement Foreground..... (EFG)

Dans l'environnement Foreground, l'utilisateur a accès à certaines commandes. Certaines autres commandes dépendent de la commande précédente exemple: on ne peut arrêter un programme FORTRAN s'il n'a pas été démarré.



Diapositive 3 Organigramme des environnements du système SIMUL

A tout moment, l'utilisateur peut, par la frappe de la commande 'HELP', demander les explications nécessaires à chaque commande.

De même à tout moment l'utilisateur peut sortir de SIMUL par la commande OUT. Il est nécessaire d'utiliser OUT, si l'on veut obtenir une bonne gestion des fichiers.

Les bibliothèques:

Le système SIMUL gère un certain nombre de bibliothèque:

1. Bibliothèques utilisateurs.
2. Bibliothèques systèmes.

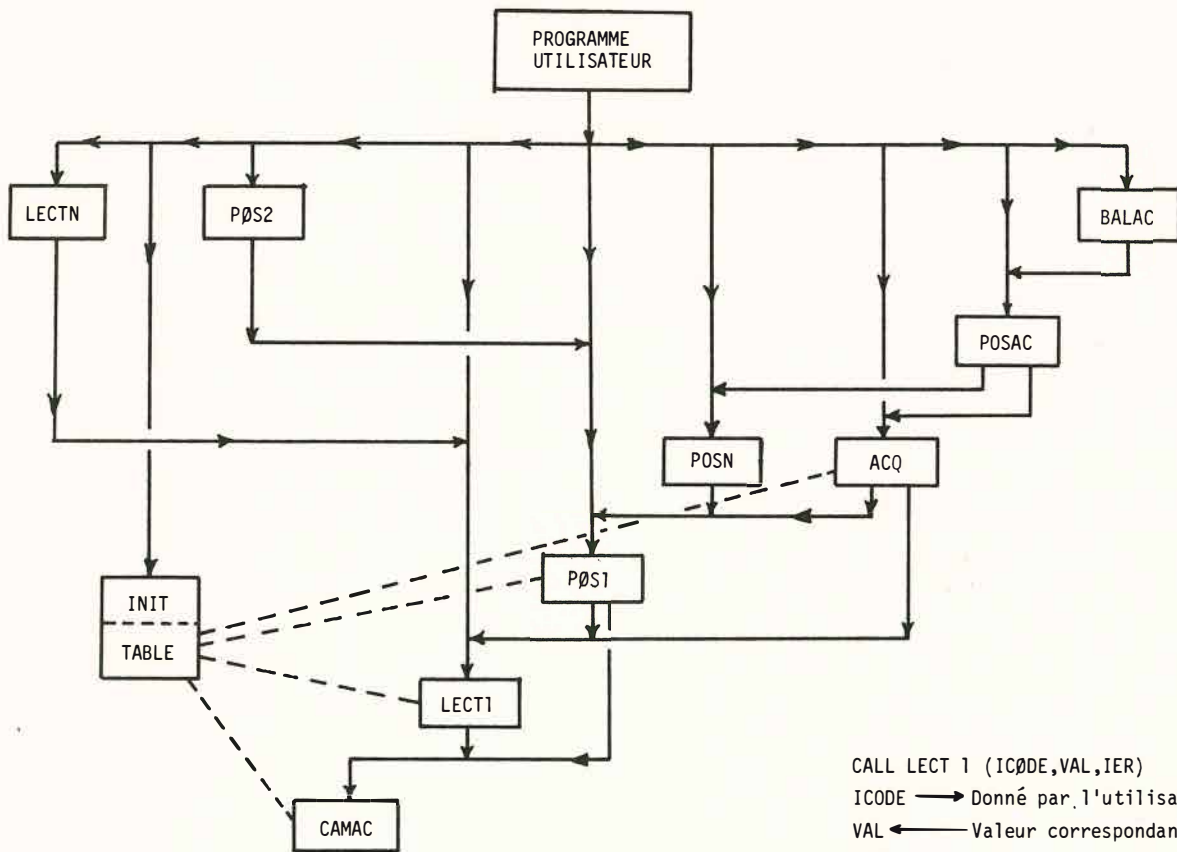
Dans les bibliothèques utilisateurs on comprend:

1. La bibliothèque de sous-programmes.
2. La bibliothèque de programmes.

Ces bibliothèques sont extensibles, et compressibles.

Les bibliothèques systèmes comprennent:

1. La bibliothèque standard FORTRAN.
2. La bibliothèque Temps Réel.
3. Les programmes systèmes.



Diapositive 4 Modules

CALL LECT 1 (ICØDE,VAL,IER)
 ICØDE → Donné par l'utilisateur
 VAL ← Valeur correspondante
 IER ← Code retour: 0 bon
 1 erreur LECT 1
 -1 erreur CAMAC

Organigramme

La bibliothèque temps réel, est celle qui représente en fait, le coeur de la simulation. C'est à travers ces sous-programmes que l'utilisateur peut avoir accès à son électronique, donc à son expérience.

Le module de base est le module CAMAC, c'est lui qui simule l'interface de transfert, tous les autres modules feront appel à lui, lorsqu'ils voudront communiquer avec l'expérience.

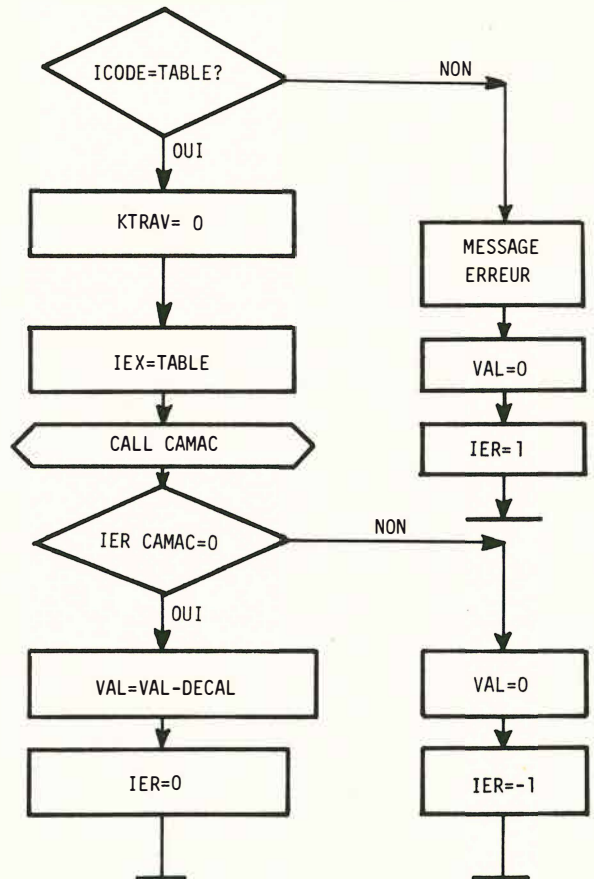
Le standard CAMAC consiste en une gestion du transfert d'information bidirectionnelle. Chaque module hardware CAMAC est repéré par une adresse (exemple COUNTER adresse BAC 1 - MODULE 15 SOUS-ADRESSE 2), pour éviter d'indiquer à chaque fois ce genre d'adresse, chaque variable physique CAMAC est repère dans le FORTRAN par un ICØDE (entier), utilisé par le module CAMAC pour rechercher la réelle adresse.

Nous avons défini 3 sortes d'action au niveau du module CAMAC:

1. Lecture d'information.
2. Ecriture d'information.
3. Fonctions spéciales telles que RAZ échelles; initialisations, etc.

Exemple d'appel du module CAMAC:

CALL CAMAC (ICØDE, KTRAV, IEX, VAL, IER).



Diapositive 5 Exemple de module LECT 1

ICODE est lié directement à la simulation, c'est lui qui fournira ou recevra l'information VAL. Dans le fonctionnement réel, ce sera le hardware lui-même qui fournira la valeur.

KTRAV indique la fonction: lecture, écriture spéciale.
IEX est l'adresse du module hardware concerné.
VAL information à transmettre.
IER code retour d'erreur 0 ----> bon
1 ----> mauvais

Les modules de plus haut niveau se composent de:

LECT 1 - LECTN

POS1 - POS2 - POSN qui font appel à LECT1 - LECTN

1V - Conclusion

Ce système de simulation fonctionne depuis le mois de Juin 1971 à l'institut Max von Laue-Paul Langevin de Grenoble, en donnant entière satisfaction.

Nous avons pu constater, que la facilité de mise en bibliothèque de sous-programmes particuliers, permettait une évolution constante du système en fonction des réalisations hardware.

De plus, ce système libère les calculateurs de processus de tâches importantes telles que édition, correction de programmes FORTRAN, en les transportant sur IBM 360/67 de fonctionnement plus souple, et aux plus larges possibilités.