COMPAGNIE INTERNATIONALE
POUR L'INFORMATIQUE

DT/DSB/MP/CO/MA-em/73-46014

Louveciennes, le 12 Janvier 1973.

# C O M M U N I C A T I O N

de M. ASSABGUI

à M. TRUPHEME

cc/ M. BOULLE
M. BONDI
M. GAUTHIER-VILARS
M. PELISSIER
M. MERCURY
M. RIGAL

OBJET : VALIDATION DE SOCRATE/IRIS 50.

Je vous prie de trouver ci-joint une première version du projet de validation de SOCRATE/IRIS 50.

M. ASSABGUI

# VALIDATION DE SOCRATE SOUS SIRIS 2

I - INTRODUCTION

II - LA VALIDATION

II.1 Validation des aspects "langage"

- II.1.a - langage de description
- II.1.b - langages de citation et de commande
- II.1.c - langage de commande - aspects conversationnel
- II.1.d - définition et appel de macro-procédures
- II.1.e - l'éditeur de textes et les utilitaires

II.2 Validation des aspects "système"

- II.2.a - intégration sous SIRIS 2 et messages de contrôle
- II.2.b - sécurité et restauration
- II.2.c - secret
- II.2.d - options déterminant la méthode d'accès
- II.2.e - interface batch
- II.2.f - définition et appel de séquences en langage d'assemblage
- II.2.g - utilisation multi-console - conflits d'accès
- II.2.h - utilisation multibase

II.3 Endurance et performances

- II.3.a - structure et données : saturation des tables
- II.3.b - contrôle donné à l'utilisateur concernant la gestion de la mémoire virtuelle
- II.3.c - charge du système en utilisations multi-base et multi-console

## III - MISE EN OEUVRE

III.1 - bases sur lesquelles sera réalisée la validation

III.2 - plan de travail et calendrier prévisible

-o-o-o-o-

# I - INTRODUCTION

La validation de SOCRATE proposée comprend trois parties :

La validation de SOCRATE en tant que langage, la validation des aspects "système" de SOCRATE et enfin un certain nombre de tests qualifiés de "tests d'endurance et de performances".

La validation de la partie "langage" comprend d'abord les tests concernant la syntaxe du langage de description. La validation du langage de description du point de vue sémantique se réalisera lors de la validation des langages de citation et de commande. La validation du langage de commande comprendra, outre le contrôle de la sémantique, celui des aspects conversationnels du langage de requêtes.

La définition et l'appel des macro procédures seront testés ensuite parallèlement. Nous avons rattaché à ce chapitre la validation de l'éditeur de textes et les autres utilitaires (un peu arbitrairement). La validation des aspects "système" comprend celle de l'intégration de SOCRATE sous SIRIS, de la gestion de la base des bases, des séquences de démarrage d'une application, de la sécurité, du secret, du détail des méthodes d'accès, de l'interface batch, des appels de séquences écrites en langage d'assemblage et enfin de l'utilisation du produit en multi-console et multi-base.

Ces deux premières parties ignorent les problèmes posés par la dimension de la base : nombre d'identificateurs de la structure et nombre de réalisation d'entités possibles. Ces deux problèmes seront examinés dans le troisième chapitre de la validation intitulé "endurance et performances". La validation des possibilités données au programmeur pour contrôler l'affectation d'espace dans la mémoire virtuelle sera réalisée ici ainsi que celle de l'utilisation en multi-base et en multi-console du système.

La dernière partie du présent projet concerne la mise en œuvre de la validation envisagée.

Cette décomposition des fonctions à tester permet d'y faire apparaître deux groupes (langage et système) pouvant être validés en parallèle et dans une certaine mesure indépendemment. La prise en charge du second nécessite une connaissance de SIRIS 2 et une appréhension globale et complète de SOCRATE. La validation des aspects "endurance" et "performances" de SOCRATE se fera dans un second temps.

Le présent document décrit les fonctions à valider dans un ordre et sous une forme qui veulent être ceux des séquences de tests : les têtes de paragraphes du plan ci-dessus pourront accompagner en leur servant de commentaires les séquences de tests.

Elles constituent par là un premier avant projet de la validation proprement dite.

## II - LA VALIDATION

### II.1 - Validation des aspects "langage"

#### II.1.a langage de description

Les tests à effectuer devront successivement porter sur des déclarations syntaxiquement correctes puis des déclarations syntaxiquement incorrectes mettant à l'épreuve l'analyseur des déclarations de structure en ce qui concerne :

- l'écriture des identificateurs
- les déclarations de caractéristiques simples de différents types
- les déclarations de type bloc et les problèmes de portée qui se posent au niveau syntaxique
- les déclarations de type entité, référence simple, référence associée à un anneau,
- les déclarations de type classe et inverse.

#### II.1.b langages de citation et de commande

Les points à tester (en interrogation et à l'aide d'une requête simple à un objet) sont les suivants :

- qualification descendante en forme préfixée à l'aide de qualificateur des types bloc, entité, anneau, inverse, classe, référence, variable xi, mot clés-xo.
- idem pour la qualification descendante en forme post-fixée.

L'on testera ensuite les points suivants :

- utilisation des qualificateurs xi (vérifier et contrarier les règles d'emploi des xi)
- utilisation des xi, yi, zi
- définition des portées de qualification (en particulier à l'intérieur des conditions)
- qualificateur xo
- filtre ayant - (vérifier et contrarier les règles d'emploi
- filtre tel que
- utilisations du point virgule
- conditions simples, syntaxe (contrôle de toutes les possibilités de comparaisons avec panachage des types de valeurs et des opérateurs).
- conditions d'existence. (Utilisation de l'indéfini u en particulier).

Pour les conditions simples et d'existence, tester les forme préfixées et post-fixées. Conditions composées (composition à l'aide de ET et OU de compositions simples)

- alors sinon
- pour la commande M vérifier le tableau des compatibilités entre parties gauche et droite
- tester les fonctions d et numde (en particulier avec des filtres)
- tester les opérateurs arithmétiques (écriture des expressions) ; en particulier du point de vue de leur priorité relative
- tester les fonctions d'édition i (édition simple et commande ECRIRE) règles d'emploi et sémantique.

- requête si, POUR et FAIRE : test de l'imbrication - test de SUIVANT et SORTIE
- requête DEPUIS

II.1.c langage de commande - aspects conversationnels

- vérifier qu'on a un dialogue chaque fois que nécessaire (selon les combinaisons type de commande - type de caractéristique)
- vérifier que la nature du dialogue est bien celle qui correspond au type d'objet cité. (ce pour tous les types)
- test de ETX
- contrôle sémantique de l'action de C, G, M, S en mode conversationnel
- requête POUR contrôlée - utilisation en conversationnel de SUIVANT et SORTIE
- requête FAIRE (ainsi que SUIVANT ou REFAIRE et SORTIE).

II.1.d définition et appel de macro-procédures

- test concernant l'écriture du nom de macro - (tester tous les cas d'invalidité est important du point de vue pratique).
- test des imbrications par l'appel (MA appelle MB qui appelle MC) comportement en cas de récursivité simple
- test des appels : dans une requête simple, dans un modèle d'appel, dans un modèle d'expansion, comme paramètre. Test de la reconnaissance des séparateurs s'ils sont identiques dans le cas d'appels imbriqués.
- questions précompilées - comportement avec différents programmes de requêtes (complets et incomplets)

II.1.e l'éditeur de textes et les utilitaires

Non rédigé pour l'instant.

II.2. Validation des aspects système

II.2.a intégration sous SIRIS 2 et messages de contrôle

- intégration sous SIRIS 2 dans une partition de 55 Ko et la configuration minimum
- mise en œuvre : gestion de la base des bases
- test des caractères spéciaux et de ATTENTION
- test de séquences (correctes et incorrectes) de commandes du 1er niveau (ATTENTION, LOGIN, PROCEED, etc...)
  test de la passation des paramètres (en une fois ou en plusieurs)
- fonction (E, D, C, M, R) test de toutes les réponses possibles en batch et en conversationnel.

II.2.b sécurité et restauration

Les tests de sécurité porteront sur une base multi-volumes. Ils consisteront surtout à définir des points de reprise, à provoquer des incidents artificiellement et à vérifier qu'à partir des deux journaux JNLA et JNLB, tant pour les reprises à chaud que pour les reprises à froid, la base est remise dans l'état où elle se trouvait lors du dernier SAVE.

II.2.c secret

- définition de listes d'utilisateurs à la déclaration de structure et des droits associés
- séquence de LOGIN avec des utilisateurs ayant des droits différents par rapport aux différentes bases déclarées.

- test en multi-console et multi-base.

II.2.d options déterminant la méthode d'accès

Les options suivantes seront testées :

- recherche sur une caractéristique rapide triée
- recherche sur une caractéristique discriminante triée
- recherche sur une caractéristique rapide non triée
- recherche sur une caractéristique discriminante non triée
- création et utilisation de caractéristiques de type inverse de différents niveaux
- création et utilisation de caractéristiques de type classe de différents niveaux.

II.2.e interface batch

(n'existe pas pour l'instant : on testera ici pour le moment les points qui suivent).

- déclarations "formal" correctes et incorrectes
- créations correspondantes.

II.2.f définition et appel de séquences en langage d'assemblage

- définition - compilation - stockage et appel d'un programm écrit en langage d'assemblage.

II.2.g utilisation multi-console - conflits d'accès

- réalisation d'opérations "courtes" (interrogations simples) et d'opérations "longues" (boucles) en multi-console

- bloquer et libérer (tests portant sur des caractéristiques utilisées en boucle pour contrôlée par exemple).

II.2.h utilisation multi-base

- réalisation de séquences de LOGIN sur plusieurs bases différentes - cas d'erreurs possibles
- appel de l'une ou de l'autre (Pb. de la sécurité et Pb. du secret).

II.3. Endurance et performances

II.3.a structure et données : saturation des tables

Le nombre total d'identificateurs est-il limité ou non ? (mise en commun des tables d'identificateurs).

- saturation des tables d'identificateurs de caractéristiques
- saturation des tables d'identificateurs de noms de macro-procédures
- saturation des tables de séparateurs de paramètres dans les appels de procédures (coupler le cas échéant les 2 tests précédents)
- saturation des noms de programmes de hash-coding
- gestion des tables de hash-code dans le cas ordonné et dans le cas non ordonné.

II.3.b contrôle donné à l'utilisateur concernant la gestion de la mémoire virtuelle

La variation des paramètres cités ci-dessous portera sur

les limites autorisées et sur les réservations qui en découlent dans l'espace virtuel :

- taille des sous-pages
- longueur de caractéristiques de type mot, valeur numérique, texte,
- nombre d'alternatives dans les listes de valeurs
- nombre de réalisations d'entités
- chaînes de bits d'inversion
- réservation globale dans l'espace virtuel d'une base comportement lors d'une saturation globale de l'espace virtuel et lors d'une saturation partielle (saturation partielle : 50 réalisations de prévues et essai de création de 51).

II.3.c charge du système en utilisation multi-base et multi-console

Pour l'instant on se contentera de vérifier que les programmes sont bien réentrants : que les mêmes phases sont activables simultanément sans conflits.

(Des statistiques sur des exploitations effectives permettront seules d'évaluer les pertes de performances entraînées par le partage de temps).

## III - MISE EN OEUVRE

### III.1. Bases sur lesquelles sera réalisée la validation

Les tests analytiques de la première partie (aspects "langage") seront réalisés sur une petite base dont les données sont introduites "à la main" - (3 arborescences et une vingtaine de réalisations pour les objets de plus haut niveau de chacune).

Les tests de la seconde partie également sauf ceux concernant la sécurité et la création ("interface batch" en II.2.e). Ces deux ensembles de tests devront porter sur des fichiers multivolumes d'une certaine importance.

Les tests de la troisième partie porteront sur des extensions de la base artificielle utilisée pour la première.

### III.2. Plan de travail et calendrier prévisible

Les deux premières parties pourront être menées en parallèle. L'on peut considérer que les points cités en II.1 et II.2 sont d'égale priorité à l'exception de la validation de l'éditeur de textes et des utilitaires qui pourra être réalisée à la suite de celle de l'ensemble des aspects "langage".

A la suite de la validation des points énumérés en II.1 et II.2, il sera procédé à l'examen des cas de saturation des tables du compilateur et de la mémoire virtuelle. La validation des aspects "langage" ne nécessitera un travail d'analyse et de programmation qu'il est possible d'évaluer en première approximation à deux hommes-mois et devrait durer environ un mois. IL en va de même de celle des aspects "système". La correction des erreurs éventuelles de celle-ci étant plus délicate sa durée pourrait toutefois être supérieure à un mois.

La partie "Endurance et performances" pourrait durer environ trois semaines les mesures de charge étant arbitrairement écourtées. (Des mesures de charge précises dureraient beaucoup plus longtemps et ne seraient significatives que dans le contexte d'une exploitation réelle).

Le calendrier de validation suivant peut être prévu :

15 Janvier - 12 Mars

- création d'une petite base permettant de valider tous les aspects "langage"
- définition de la méthode de validation de chacune des fonctions de SOCRATE
- écriture des programmes de validation pour un certain nombre de ces fonctions

12 Mars - 15 Avril (volume de travail ~4 hommes-mois)

- validation des aspects "langage"
- validation des aspects "système"

} en parallèle

15 Avril - 1er Mai (volume de travail ~2 hommes-mois)

- "endurance et performance"

1er Mai - Fin Mai (volume de travail ~4 hommes-mois)

- tests supplémentaires (rebouclage total ou partiel de la validation) consécutifs à la correction des erreurs de la version du 12 Mars

La fin de la validation peut donc être approximativement prévue pour la mi-Juin.

M. ASSABGUI