PS/Co Min 85/43

C O M P T E R E N D U D E R E U N I O N :

Environnement software du SMACC (dans le cadre des NAPS)
des 18 et 25 Juin 1985

Cl.H.Sicard.

PRESENTS:G.Benincasa, R.Cailliau, L.Casalegno, G.Cuisinier, A.Daneels, N.de Metz Noblat, F.Di Maio, A.Gagnaire, F. Giudici, W.Heinze, P.Heymans, B.Kuiper, J.Lewis, E.Malandain, W.Remmer, C.Serre, P.Skarek, G.Shering, F.Perriollat, C.H.Sicard.

Objectif: Examiner l'environnement software nécessaire pour la 1ere utilisation du SMACC par les NAPS.
Le tour de table devrait faire apparaitre comment les besoins s'ajustent aux outils prévus, et aussi les extensions et compléments d'outils nécessaires.
Les SMACCs "autonomes" (IK-Box, Caviar, Linac) ne sont pas inclus.

Points abordés:

1) Organisation des applications dans le SMACC. (Cl.H.Sicard)

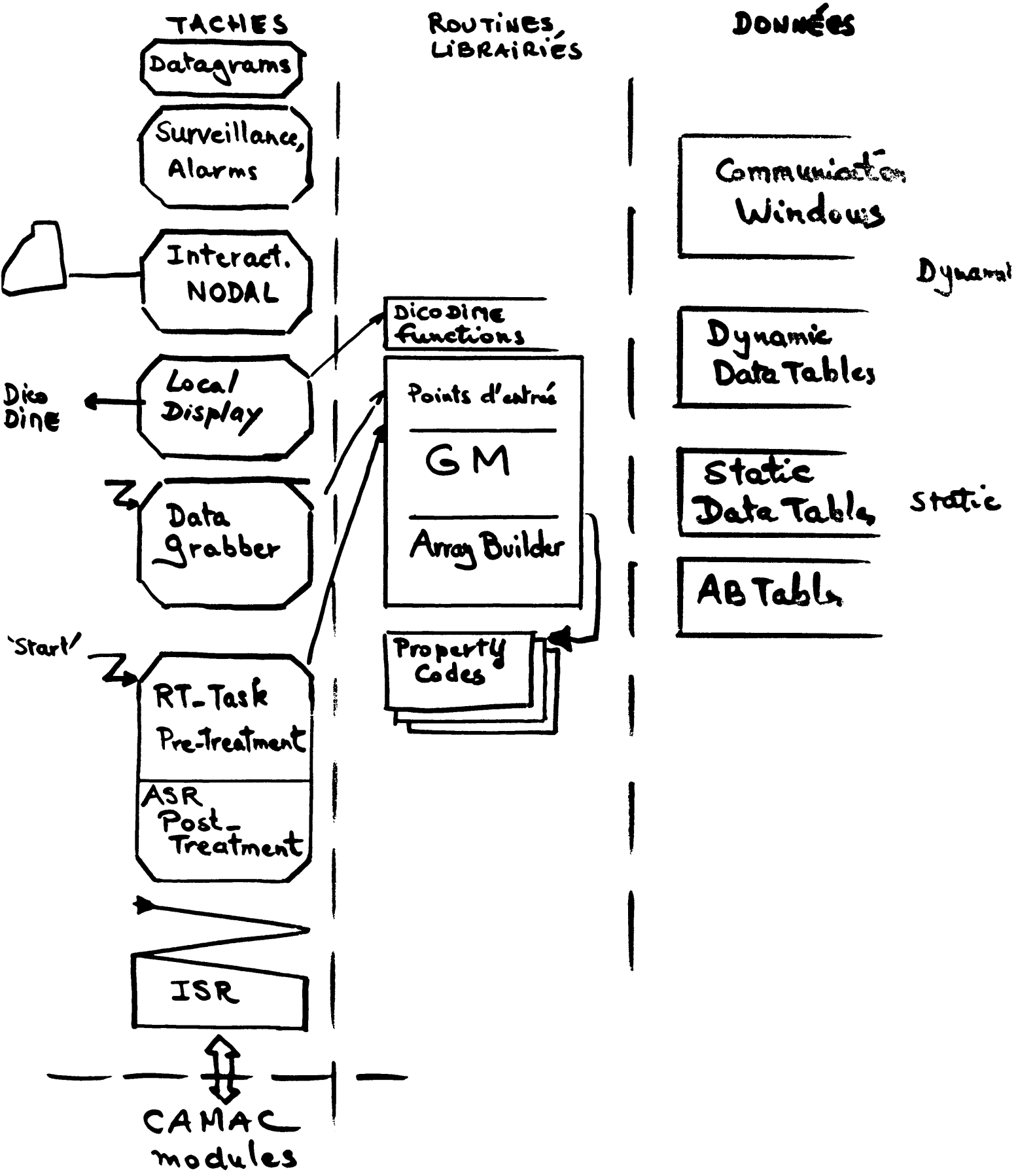
- Repartition de l'activité entre taches, routines, ISR.

- communication, synchronisation (interne SMACC, avec le FEC)

- programmes géeéraux de service.

Voir figure;
En résumé:
-3 interrupts sufisent à déclencher les taches RT:
"start" pour préparer les tables à transmettre chaque cycle.
"ISR" pour transfert de/vers CAMAC.
"MDR" pour declencher le ramassage des données. Cette tache peut ensuite en declencher d'autres (local display,..)

ORGANISATION des activités AP dans le SMACC



Questions posées:

a) Est-ce utile et possible d'avoir d'autres Nodal que l'interactif?

b) Veut-on travailler avec le Nodal interactif depuis le FEC?

c) Comment établir les connections (Mac-)Terminal <-> SMACC? proposition d'un patch-panel près de la console en EB1.

d) Quels diagnostics locaux?

2) Nodal (G.Cuisinier)

- fonctions systeme, addition de fonctions user.

- Nodal interactif, autres Nodals non interactifs.

- SYS-GO

Presentation:
Nodal Interactif: diffère du RT-Nodal FEC par:
-fonctions definies allouées dynamiquement;
-minuscules acceptées;
-extension des commandes d'edition;
-fonctions type"Basic": Data, Read, Restore.
-conformation au standard Cern.
-ajout du "scratch-file" (ds working-area) -utile pour tests
-fonctions d'accès au File-Module (64K) Camac.
-Entrées/Sorties: accès MacIntosh (Imprimante)
-Commandes RMS : précédées de @
-priorité Nodal modifiable (comme autres taches).

Autres Nodals: Serveur IMEX et EXEC installés,
mais sans la communication FEC.

SYS-GO: RMS au startup active la tache "Monitor" unique qui déclenche les 3 autres taches Nodal (Int,IMex,EXec).
Il n'est pas prévu de SYS-GO "Application", variable d'un système à un autre.(progr Nodal, mais stocké ou?)

Questions:

a) peut-on passer interactivement de Nodal à Monica?
Oui,moyennant qq modifs dans Nodal et Monica.

b) Chargement de code machine dans fonctions definies?
Non (pas pour l'instant).

c) fonction EWAIT demandée (attente d'un interrupt Front-Panel ou PLS-Receiver)?
? Pas inclus; l'emulation semaphore pour chaque
LAM coute cher; risque qu'une tache crashe sans faire le clear LAM.

d) Comment Nodal connait-il les fonctions "applic"?
Il cherche à une adresse magique la première fonction, a laquelle les autres doivent etre chainées.

3) P-Plus (R.Cailliau)

- facilités version B prévues.
Toutes celles dans la definition sont prévues, les plus importantes seront fournies en premier.

- Run-time et I/O (Camac,terminal, fichier)
Les variables globales (au sens librairie Nord-100) sont remplacées par les primitives de RMS (Event Messages).
Toute fonction P+ rendue visible à l'extérieur est génerée avec son header Nodal. (il n'y a pas de code spécifique necessaire pour rendre une fonction ICCI, puisque la convention d'appel est la meme sur le SMACC.
Le Camac sera accédé comme un accès mémoire, via un tableau indicé (L'adresse de base Camac étant definie globalement).
Rien n'est prévu pour les fichiers ni les entrées-sorties Terminal; pour ce dernier, le seul moyen serait de communiquer par des messages avec un programme special du Nodal interactif. On ne peut donc pas faire de trace directement depuis un programme P+.

- debugging (Monica et interaction)
Un debugging style assembleur sera possible, a l'aide du listing assembleur fourni par le compilateur. Il existe dans le Nodal-68000 un debugger simple (break/registres/memoire).
Pour mettre en oeuvre MoniCa, il faut (outre suffisament de mémoire SMACC) une géneration des descripteurs P+ pour le code et les données. Ce dernier point n'est pris en charge par personne.

4) Librairie generale (R.Cailliau)

equivalent de process-manipulation;
Pour les activations de taches, il faut choisir entre une interface directe vers les Monitor-calls de RMS, ou une mise en oeuvre de 'Process-Manipulation' qui serait plus portable (mais plus longue à mettre en oeuvre.) Au vu des contraintes de temps, la première solution sera choisie (en tout cas dans un premier temps).

contexte des programmes utilisateurs.
R.C. demande un document décrivant les éléments nécessaires à l'integration des programmes Utilisateurs, comportant les points suivants:

a) Layout mémoire (RAM/EPROM), Taches.

b) Initialisation dynamique.

c) Contexte vu par le programme utilisateur.

N.M.N rassemblera ces informations. A cette occasion, il faudra préciser certaines options (comment créer les données statiques, stack-area, adresses magiques necessaires à la communication FEC,...)

5) Loader (G.Cuisinier et F.Di Maio)

- Chargement depuis FEC

L'image est sous forme binaire dans le FEC, avec des enregistrements sous la forme (Load-adress/ Word-count/ Data(<256)). Elle est transmise au SMACC par LDACC qui effectue la séquence:

Stop-ACC PUTBL Start-ACC.

Un Task-builder a été ecrit par G.C pour recharger dynamiquement une tache (en code 'position-independent') via un serveur dans le SMACC. cela pourrait en principe etre utilisé pour du code P+, qui est position-independant , sauf pour l'initialisation du stack au démarrage.

L'allocation des segments de données (fait dynamiquement) peut poser un problème en cas de chargement partiel de taches.

- chargement depuis MacIntosh

Il n'est pas prévu de charger les images des SMACCs depuis le McIntosh, dans la mesure ou un FEC devrait etre disponible pour cela.

6) Interaction locale et Stand Alone (F.Di Maio)

- MacIntosh terminal, file serveur, graphic. Le McIntosh remplit les fonctions de terminal SMACC ainsi que de serveur fichiers (LOAD ou SAVE de fichiers Nodal, et aussi accès fichier direct depuis les programmes Nodal (mais pas P+?) (OPEN,READ,WRITE).

- Graphic local (Dicodime et emulateur)

Une librairie de routines d'accès aux fonctions de base (Posit, Write) puis graphiques (Histogrammes) sera fournie par A.G/ F.d.M.

- facilité de transfert PRDEV-MacIntosh

C'est possible via le PACX, en connectant le McIntosh comme un terminal du PRDEV.

7) Init et Start-Up (Ch. Serre)

- chargement et sauvegarde des données Il est rappelé que les Data-Tables des Modules sont maintenant partagées en une partie 'Read-Only' (rechargée par LDACC) contenant les adresses Camac, les Min,Max... et une partie 'Operationnelle' modifiable en ligne; un programme local ne pouvant opérer que sur une copie 'locale' de cette table.
  La sauvegarde des données se fait pour les *données* operationnelles sur le disque FEC, soit au Release soit toutes les quelques minutes.

- démarrage des protocoles (communication).
  Lors d'une redémarrage d'un SMACC, un echange de messages doit etre fait avec une tache dans le FEC pour reinitialiser les protocoles de communication.

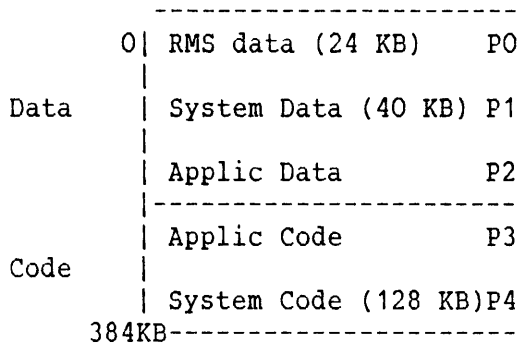
8) RMS (W.Remmer)

RMS version 4.3 et MIOS version 1.0 sont disponibles

- Ressources (TCB, semaphores,partition memoire,division supervisor/ user, lignes terminal allouées pour interaction et trace)

Le chargement de la mémoire se fera en 2 fichiers séparés (un Système, un Applic).

l'allocation mémoire prévue est la suivante:



En résumé, il reste environ 192KB pour les applications. C.H.S fera une nouvelle estimation de la place mémoire nécessaire pour les NAPS. (hypothèse de 50% d'expansion de code par rapport au Nord).

La partition 'Applic Data' est initialement vide; l'allocation de segments de données est faite par les programmes, à l'initialisation.

La frontière entre P2 et P3 pourrait etre dynamique.(C'est demandé par les AP car les dimensions des tables peuvent varier largement d'un SMACC à un autre.)

Le code système contient les I/O Drivers, Nodal et P+. Il n'inclut pas MoniCa, qui prend de 64 à 100 KB. Dans le cas d'un système en RAM, cela ne laisserait que 92KB (ou 48kw) pour les applications, ce qui est insuffisant. Par conséquent, il est illusoire de compter sur MoniCa (ce qui est dommage, à la fois par l'aspect debug symbolique qu'il offre, et aussi par l'effort investi pour s'y adapter.)

La répartition est faite dans l'hypothèse de 16 taches; Il faut compter 1/2 KB par Task-Control-Block (TCB).

Les autres ressources (semaphores,...) prennent peu de place et peuvent donc etre etendues si besoin est.

La mise sur EPROM des noyaux de code RMS et Nodal pourraient faire gagner 64KB; ce n'est pas à prévoir pour l'instant. D'autre part, pour l'exploitation il faut des SMACCs standards (meme configuration mémoire et EPROM) pour interchangeabilité.

- Traitement des erreurs (trace, action) Si les taches sont construites sous forme d'un 'Subtask' d'un Monitor Task, la tache Monitor recoit les erreurs fatales de ses Subtasks et peut les réactiver. Ceci est particulièrement important dans le cas d'une 'Bus-error' pouvant provenir par exemple d'un accès Camac en Read au lieu de Write.
  Une autre facon de se protéger est d'utiliser la directive 'Announce Exception Vector'.

  Une facilité de trace est disponible sous RMS, qui trace les appels aux directives système/ interrupts/ Bus errors... dans une table qui peut ensuite etre dumpée à un moment opportun. Cela ralentit le système.

  Il est demandé à W.Remmer d'organiser une réunion sur le sujet 'comment utiliser les directives RMS et les I/O drivers'.

9) Planning.

a) P+

On espère que début Septembre des personnes 'expérimentées' pourront commencer à produire des applications.

Pour arriver à réaliser cet objectif, cela necessite que R.C. se consacre exclusivement à P+; les autres activités de la section système sont partagées entre les autres membres de la section (voir memo de B.K.)

b) Datagrams et Remote-Procedure-Call.

Un prototype de Datagrams existe entre N100s, amélioré par rapport à la version initiale (performances, trace). Le prototype pèche encore par un manque de ressources offertes et des défauts de synchronisation. Une investigation de XMSG est proposée pour apporter une solution propre à ces problèmes.

Pour la disponibilité du RPC entre FEC et SMACC, il est prévu entre 3 à 4 mois (soit le 1er Novembre, en supposant 2 mois restants après la disponibilité de P+).

Une réunion est à prévoir (F.P.) pour examiner l'aspect technique. Un mécanisme spécifique pour l'appel FEC-> GM SMACC est envisageable comme solution de recours si la date visée n'etait pas atteignable.

c) IKBox et TSU. Les choix à faire pour ces SMACCs spéciaux necessitent une réunion (demandée par G.B. et Ch.S pour IKBox) afin de définir leur layout et le software à realiser.