Eurogam project **EDOC118**

Graphics developments

Principles and architecture

---

edition : 1.0
date : 15 february 1991
auteurs: G.Michel & G.Zehnacker

Centre Recherches Nucleaires Strasbourg
CNRS-IN2P3 France

Graphics developments

L.Michel & G.Zehnacker

Edition 1.0 15 february 1991

Abstract

Après la rencontre avec nos collègues de Daresbury (15-16-17 janvier) à Strasbourg, nous avons mis en forme et approfondit les idèes concernant le graphique .

Dans le cadre de la collaboration Eurogam seront developpées de nouveaux logiciels notamment graphiques. Ceux ci devront être construits avec les mêmes idées de base et comporter un 'look' identique à Daresbury et Strasbourg de telle façon que le physicien passe sans difficulté d'une interface graphique à l'autre .

1 Introduction

1.1 Pourquoi un nouveau logiciel graphique ?

Les laboratoires du NSF de Daresbury et du CRN de Strasbourg dispose actuellement de logiciels graphiques qui doivent être revus pour satisfaire aux exigences du multidétecteur Eurogam et des demandes précises des physiciens. A Strasbourg , les nouveaux logiciels doivent aussi satisfaire aux demandes des autres acquisitions auprès du Vivitron avec Demon, Icare et les Multicompteurs au Q3D.

1.2 Ecran de présentation

Au démarrage l'écran se présentera sous la forme de la figure 1 où plusieurs logiciels seront disponibles aux physiciens, Il suffira de cliquer dans une des cases pour le selectionner. (**voir figure 1**)

2 Présentation de l'architecture

2.1 Quelques principes

Nous allons préciser les principes suivis pour concevoir le nouveau logiciel graphique qui sont :

- de développer sur des stations de travail connectées en réseau ethernet
- de dialoguer avec les logciels par clavier et souris
- d'adapter les logiciels pour qu'ils soient appelables à partir par des terminaux classiques (type tektro ou falco)
- de tenir compte des nouveaux outils de developpement disponibles (notamment Dev guide sur SUN voir EDOC116)
- de tenir compte des bibliothèques graphiques (xlib sous X-Window version x11R4) pour tracer les spectres
- d'utiliser des bibliothèques appelées toolkit intermédiaires entre la xlib et l'application pour créér boites de dialogue , boutons , menus etc
- de permettre à l'utilisateur d'adapter lui même le soft à ses besoins (avec le moins de recompilation possible)
- de tenir compte des possibilités des nouveaux matériels et logiciels de base .
- de réutiliser les fonctions d'analyse qui ne changent pas

2.2 Cadre de l'étude

Une première proposition d'interface graphique avec quelques fonctions d'analyse a été proposée dans EDOC112. Une approche d'architecture interne est présentée dans EDOC110. Dans le présent article nous en tiendrons compte pour poursuivre nos reflexions et concevoir une architecture interne du nouveau logiciel graphique (**voir figure 2**). Celle ci sera trés modulaire et sera alors plus facile à développer et à maintenir.

2.3 Schéma général

Les commandes demandées sont sélectionnées par l'utilisateur depuis son terminal ou de sa station, puis mise en forme pour être envoyées à un shell qui sèlectionnera la fonction à activer à partir de différentes librairies .

La figure 2 montre à gauche les différentes façons de saisir une commande UNIX:

- soit par l'intermédiaire d'une fenêtre shell tool sous Open Windows
- soit par un shell sur une console Tektro 4014 (par exemple)
- soit par un interface graphique par l'intermediaire de menu
- soit par un shell sur une console X

Au centre du schéma un shell recevra ces commandes et activera des programmes d'analyse provenant de différentes librairies à gauche.

2.4 Différents niveaux dans le logiciel

Dans notre analyse nous aurons à l'esprit les principes décrits plus haut et les concepts X-Window décrits dans l'annexe 1. En allant donc plus au fond des choses, nous décomposerons le premier schéma proposé en plusieurs niveaux (**voir figure 3**) . Ceux ci sont dans l'ordre à partir d'une action sur le terminal :

- un driver de terminal spécifique à chaque station ou terminal
- un serveur X
- des progammes clients
  - Xterm,
  - Window manager,
  - analyse de spectres,
  - setup,
  - acquisition,
  - controle (monitoring)
  - test de carte electronique

  - analyse de spectres,
  - selecteur de logiciel (voir figure 1)

- des serveurs fonctionnels
  - serveur de spectres
  - shell
  - autres

- des accés à des ensembles d'informations sous forme
  - de librairies de fonctions d'analyse,
  - de base de donnees,
  - de fichiers de spectres

3 Le X server

Ce module est un produit fourni avec tout le logiciel X-Windows du MIT. Nous n'aurons pas à le modifier. Il prendra en compte toutes les actions faites à la station (voir annexe 1) et les transmettra à un client. D'un autre coté, il exécutera sur l'écran de la station toutes les requêtes reçues par exemple soit du shell, soit de l'afficheur de spectres ou d'autres clients (notamment du window Manager) .

4 Différentes sortes de clients

4.1 Xterm

C'est lui qui recevra les commandes UNIX . Elles seront transmises à un shell (Bourne shell ou C-Shell) qui lancera une commande shell, un programme utilisateur ou un script. La tache créée par le shell pourra être un programme d'analyse (affichage de données, intégration, fit, recherche de pics, etc) ou bien une action sur une fenêtre de spectres.

4.2 Window manager

Le window manager est un client chargé de la gestion des écran des clients. Ce client prend en charge toutes les fonctions de réorganisation de fenêtres (ouvrir, fermer, agrandir, faire disparaitre les fenetres , etc). Ce client tourne sur une machine du réseau. Il a en charge l'aspect visuel des fenêtres. Il peut avoir des liens avec d'autres clients. Il a en charge la gestion des menus apparaissant sur la fenêtre racine (menu en fond d'ecran)

4.3 Setup

De nombreuses informations sont initialisées dans un premier temps, puis modifier ou lister . Ce client serait chargé de cette fonction de "data base". Naturellement elle ne sera pas accessible de n'importe ou par n'importe qui . Un systéme de droit réservè est à mettre en place . **(voir figure 5)**

4.4 Acquisition

Pour paramétrer une experience, la démarrer, l'interrompre ou l'arrêter nous allons développer un interface graphique s'intégrant dans l'ensemble . Ce client disposera aussi de fonctions de contrôle spécifique graphique d'affichage de spectres . **(voir EDOC111 et figure 6)**

4.5 Contrôle

Les dispositifs expérimentaux doivent être surveillés en permanence et détecter immédiatement toute panne ou défaillance partielle du materiel. Un ou plusieurs clients X assumeront cette fonction. **(voir figure 7)**

4.6 Test

Ce logiciel permettra d'afficher des synoptiques de cartes (par ex GE, BGO ou trigger, etc), d'indiquer une connexion entre un composant et une des 4 entrées d'un oscilloscope externe. Il sera aussi possible de visualiser le spectre d'une voie et de modifier èventuellement quelques paramètres fournis à des composants electroniques. Un EDOC est en cours de redaction. **(voir figure 8)**

4.7 L'analyseur des spectres

Il aura en charge toutes les fonctions nécessaires pour gérer l'espace d'affichage des spectres, de prendre en charge toutes les commandes "Scope" pour parametrer le dessin à l'ecran , dessiner les spectres et d'activer les fonctions d'analyse . **voir Edoc 112 et figure 9**

4.8 Le client "menu"

Il recevra du X server un évenement préçisant fonctionnellement le nom de l'option selectionnée par la souris. Cette option devra éventuellement être completée par une saisie de données par boite de dialogue. Après mise en forme de ces renseignements sous forme d'un message , ils seront transmis vers un programme chargé d'executer les commandes.

5 Serveur de fonctions

5.1 Le serveur de fonctions d'analyse

ce module est en fait un shell UNIX qui sera appelé par le client menu qui lui fournira le nom du programme et de ces paramètres. Le dialogue s'etablit alors directement entre programme fonction et client menu.

5.1.1 Le serveur de spectres

Ce module est chargé de rechercher les spectres sur le réseau pour le compte d'un client qui pourra être par exemple l'afficheur de spectres ou un module appelé par le shell. La demande de spectre(s) se fera à travers une boite de dialogue apparaissant à l'écran qui demandera le nom de la machine, du répertoire et des fichiers spectres ou du numero du device bande. Cette information permet au serveur de rechercher le spectre au bon endroit et de le faire parvenir au demandeur de spectres.

6 Les librairies

6.1 De fonctions

Cette librairie contient toutes les fonctions d'analyse developpées au cours des années et qui seront réutilisables. La contrainte est qu'il faut y toucher le moins possible. Il faut donc respecter ce que demandent ces fonctions en entrée et ce qu'elles restituent en sortie. Nous imaginerons donc une boite de dialogue aussi universelle que possible pour l'entrée des donnees et mini fenêtre d'affichage. Cela permettra donc une entrée de commandes parametrées beaucoup plus conviviales qu'avec un Bourne shell ou un C-shell. L'appel de fonctions par menu devra suivre le même mécanisme.

6.2 De données (data base)

Les études sont en cours pour soit choisir un produit disponible(free share) soit en developper un specifiquement pour Eurogam.

6.3 De spectres

Un logiciel d'accés fournira les fonctions d'écriture, de lecture et de recherche de spectres. Nous rappelons que les spectres seront stockés soit en mémoire de l'histogrammer, soit sur une bande exabyte ou un disque situé sur une station du réseau.

7 Description du protocole entre modules

La **figure** 4 montre qu'un client pourra être en relation avec un certain nombre de modules. Sous le terme module nous entendrons à la fois les progammes clients (au sens de X-Window) ainsi que les serveurs de fonctions. Tous ces modules vont avoir à communiquer entre eux. Ces informations seront de type tres différent par exemple

- des appels à des programmes avec une liste de parametres
- des informations reçues de ces programmes qui sont à afficher
- des positions de curseur sur l'écran
- des positions de canal selectionnés sur un spectre
- des tableaux de chaines de caracteres par ex des listes de noms de spectres
- des positions de pics à materialiser avec leurs caracteristiques
- des actions demandees pour modifier l'affichage des spectres
- des modifications des fenêtres
- etc

7.1 appel au librairies externes

Le projet a été dès le début defini comme pouvant utiliser des fonctions externes appelées à partir d'un noyau graphique figé. Il a été convenu que la connection d'une librairie externe au logiciel graphique devait pouvoir se faire sans recompiler ce dernier. Par ailleurs, les fonctions externes doivent pouvoir etre utilisées dans d'autres contextes que nos logiciels graphiques.

La prise en compte d'une fonction externe peut se décomposer en quatre étapes:

- Paramétrage de la fonction (procédure interne au logiciel graphique).
- Appel de la fonction (utilisation du système d'exploitation).
- Retour des résulats (utilisation du systeme d'exploitation).
- Prise en compte des résultats (procedure interne au logiciel graphique).

A chacune de ces étapes correspond une procédure particulière developpée dans un EDOC11x en cours de rédaction..

Il nous est apparu nécessaire de pouvoir appeler ces procédures externes à partir du shell (ligne de commande). Cela impose aux fonctions externes de se présenter comme des fichiers exécutables. Une modification d'un de ces executables n'entrainera aucune modification au logiciel graphique d'analyse.

7.2 Paramétrage des fonctions

Chaque fonction externe demande un paramétrage propre. Ce dernier est realisé à partir du noyau graphique qui doit donc être capable de saisir des arguments divers sans être recompilé. La solution la plus simple consisterait à demander à l'utilisateur de taper directement la ligne de commande. Cette solution pour simple qu'elle soit, n'est pas satisfaisante car, d'une part, elle demande à l'utilisateur la connaissance d'une syntaxe et d'autre part, elle nous fait perdre le bénéfice de l'interface graphique. Nous préferons nous reporter vers l'utilisation de boites de dialogue parametrables.

Ces boites de dialogue sont créées dynamiquement à partir d'objets actifs tels que des boutons ou des éditeurs de ligne de texte. Il est donc aisé d'associer à chaque fonction externe la boite de dialogue utilisée pour son paramétrage. Le paramétrage de la fonction se fait en associant un paramètre à chaque élement actif de la boite de dialogue, puis en définissant l'ordre des paramètres à passer de maniere à pouvoir merger la ligne de commande.

Ces informations devrons se trouver dans un fichier, lu par le noyau graphique au moment de son dèmarrage. De la sorte, il suffira de créér un fichier pour chaque librairie.

7.3 Appel des fonctions

Le noyau graphique doit créér le processus correspondant à l'execution de la fonction. Ce processus voit sa sortie redirigée vers le noyau graphique afin de lui transmettre le resultat de ses calculs. Pour ce faire, nous proposons d'utiliser les 'pipes' Unix.

7.4 Retour des résultats

Les fonctions externes se présentent sous la forme de fichiers executables, leurs résultats sont dirigés vers la sortie standard (ecran).

Dans le cas d'un appel à partir du shell, le résultat s'affiche à l'ecran. Dans le cas d'un appel à partir d'un noyau graphique, il faut pouvoir rediriger les résultats vers le noyau sans modifier la fonction. Cette redirection est mise en place en créant un 'PIPE' de la fonction vers le noyau appelant au moment de la creation du processus.

7.5 Prise en compte des resultats

Les résultats arrivent par le 'PIPE' Unix sous forme de chaines de caractéres. Ils doivent nécessairement être pris en compte par une fonction interne du noyau graphique. Cela ne pose pas de problème tant que les résultats sont des valeurs numériques à afficher telles quelles. Par contre, il en va tout autrement des lors qu'ils s'agit d'interpréter graphiquement ces résultats sous forme de spectre ou de courbe de fit par exemple.

Le noyau graphique doit posséder un certain nombre de fonctions internes d'interprétation graphique des résultats acceptant comme argument les diverses formes de code de retour des fonctions externes.

Comme il n'est pas possible de prévoir tous les cas a priori, il faut proposer une liste de formats de résultats reconnus par le noyau graphique et associer à la definition de chaque appel le format idoine.

Cette liste de format évoluera au cours du temps de maniére à prendre en compte de nouvelles fonctions.

7.6 Configuration du noyau graphique au démarrage

Toutes les données décrivant les appels aux fonctions externes sont groupées dans un fichier ASCII lié à la bibliotheque concernèe. Au démarrage, le noyau graphique lit ce fichier, rempli les menus avec les fonctions declarées, crée les boites de dialogue correspondantes et positionne les drapeaux de séléction des fonction d'interpretation des resultats.

8 Listes des fonctions d'analyse

Une liste de fonctions a été établie dans EDOC112. Nous examinerons également toutes les fonctions du NSF de Daresbury. Notre objectif est de permettre à tout physicien de modifier ou de rajouter une fonction en ayant à changer le moins de choses possibles, notamment dans l'intégration dans les menus déroulants et de la modification au niveau du dialogue pour l'entrée de données.

9 Outils à notre disposition

Nous avons etudié la documentation de SUN notamment le DEV Guide pour pouvoir en dégager les facilites, les possibilites, et en apprécier mieux les limites et les adaptations à faire pour le developpement de notre produit. Ce produit Dev Guide nous a déjà permis de générer le proto de l'écran d'analyse (voir edoc112) d'une façon interactive sans programmation. Nous avons aussi déjà élaboré une méthode pour utiliser des boites de dialogue , en étant le plus independant possible des programmes d'analyse actuels qui devront etre reutilisés. Dans une deuxième phase , nous allons etudié une procédure permettant à l'utilisateur de rajouter des options aux menus d'analyse.

10 Echelonnement des travaux

Cette première approche est à discuter pour permettre de d'egager tout ce qu'il est possible de développer et tester séparemment . Le test de l'ensemble pouvant se faire ulterieurement .

11 Conclusions

Cet article fait donc un survol de l'architecture des différents logiciels utilisant un interface graphique. Notre objectif est de disposer d'un document pour l'acquisition (edoc111), le setup (en cours d'élaboration), le test de carte (en cours d'élaboration, le controle (en cours d'élaboration) et l'analyse de spectres(edoc112).

Bibliographie

Les documents concernant le graphique sont disponibles sur l'info-server
Auteurs : L.Michel & G.Zehnacker

- EDOC110 : Developpement graphique - présentation générale
- EDOC111 : Graphics developpements - Acquisition Spectra and events
- EDOC112 : Developpement graphique - Analysis Functions & Ergonomics
- EDOC116 : Devguide: a graphical interface editor
- Edoc1xx : Principles and architecture
- Edoc1xx : Display modules - analysis modules
- Edoc1xx : Setup (en cours)
- Edoc1xx : Test (en cours)
- Edoc1xx : Controle (en cours)

Appendix

Les concepts de base sous X-Window

Chaque terminal est doté d'un clavier, d'un dispositif de pointage et d'un ou plusieurs écrans. La gestion de ce materiel est assuree par le Serveur X_Window. L'application graphique est client du serveur X.

1 Communication client serveur

Le protocole est basé sur le modèle client serveur X developpé au MIT pour les interfaces graphiques modernes. Dans ce modèle les échanges entre client et serveur X se font par requêtes et X events. Le ou les clients envoient toutes les informations necessaires au serveur X pour executer des actions á l'écran . Le graphisme est imposé par le client. Le serveur X à son tour renverra une reponse suivant le type de requete du client. Le client et le server ne sont pas executes obligatoirement sur la même machine. Notons aussi que plusieurs clients peuvent interagir avec un serveur X . De la même façon un client peut communiquer avec plusieurs serveurs.

2 La fenêtre

L'objet de base avec lequel travaille le serveur est la fenetre. Une fenetre est simplement une zone rectangulaire de l'écran à laquelle est associé une application. La fenetre associée au serveur lui même est l'écran dans sa totalité. C'est la fenetre racine. La fenêtre active est normalement celle dans laquelle se trouvait le pointeur de la souris au moment ou l'evenement s'est produit. Chaque client possède au moins une fenêtre, pas forcément visible, qui lui est directement associé.

3 Les événements

L'événements traduit une action détectée par le serveur concernant un client telle une pression sur un bouton de la souris, une action sur le clavier ou encore une modification de l'emplilement des fenetres. Une caractéristique de X-Window est le traitement asynchrone des événements. Tous les événements seront empilés par le serveur et transmis au client. L'evenement est enregistré par le client sous forme d'une structure C dont le contenu depend du type d'evenement.

Un evenement pour un client peut être de differentes formes , par ex :

- la position du curseur au moment du clic,
- la transmission d'un caractere ASCII frappé au clavier
- la demande de changement de taille d'une fenêtre
- la mise en avant ou en arrierre d'une fenêtre

4 Le client

Au départ le client doit créér une ou plusieurs fenêtres de maniére à disposer d'un espace d'affichage et à être identifiable par le serveur. Puis le client est mis en sommeil dans l'attente d'evenement. L'arrivée d'un evenement génère une interruption qui reveille le client (fonction XNextEvent de la X-Lib) et executera la fonction correspondante à ce type d'evenement.