LABVIEW INTEGRE ET ACCELERE LES PROCESSUS DE DEVELOPPEMENT EN PHYSIQUE EXPERIMENTALE

O. Zimmermann, P. Stassi

LPSC / IN2P3
53, avenue des Martyrs
38026 Grenoble
{ ozimmer, stassi }@lpsc.in2p3.fr

Résumé : La physique expérimentale a massivement recours à la programmation d'instruments. Nous présenterons deux réalisations LabVIEW du LPSC, pour les expériences GRAAL et PLANCK. En nous appuyant sur ces exemples nous dégagerons certaines qualités qui permettent à LabVIEW d'intégrer l'environnement et les méthodes de la recherche. Enfin nous examinerons des points de concordance profonds entre la technologie LabVIEW et les contraintes techniques de la programmation des expériences.

Mots clés : recherche, expérimentation, langage G, programmation réactive, temps réel, machines à états finis

1 INTRODUCTION

Les laboratoires de recherche contribuent activement à améliorer les technologies qu'ils utilisent, et qui sont souvent nées de leur activité. Ce *processus bouclé de développement* à grande échelle, se retrouve à petite échelle dans le déroulement des projets de recherche expérimentale. En effet, les réalisations techniques et les résultats expérimentaux se renvoient continuellement la balle pour avancer dans la compréhension et dans la maîtrise des sujets d'étude.

Dans le domaine de la programmation, les laboratoires utilisateurs de la technologie supportent le poids d'une culture spécifique tenace : celle du calcul scientifique. Pourtant les applications de commande de montage ont un objet et un fonctionnement extrêmement lointain de ceux du calcul mathématique, et qui relève bien davantage du paradigme plus jeune de la *programmation réactive*.

2 CONTEXTE ET EXEMPLES

2.1 Des projets d'envergure très variable

L'époque contemporaine se caractérise, dans l'histoire de la recherche scientifique, par l'apparition et la multiplication de projets de très grande envergure (accélérateurs de particules, détecteurs de cosmiques de haute énergie, spatial, séquençage génomique…). Ces projets en collaborations nationales ou internationales poussent les laboratoires à s'adapter à des modèles d'organisation et des méthodes de travail différentes.

Toutefois, cette évolution n'a fait qu'étendre un peu plus une gamme de projets qui commence et commencera longtemps encore par la "manip" de table, réponse immédiate à une idée ou à une proposition d'amélioration de l'expérimentateur. Ce dernier est généralement un chercheur du laboratoire,

mais souvent aussi un étudiant ou un stagiaire.

Depuis 1998, le Service Détecteurs et Instrumentation du Laboratoire de Physique Subatomique et de Cosmologie (LPSC) de Grenoble a pris en charge des projets qui reflètent cette diversité. La programmation des montages intervient dans la plupart de ses réalisations et utilise LabVIEW et le langage C.

2.2 L'exemple de GRAAL

GRAAL est une expérience de physique hadronique en place depuis 1995 sur l'une des lignes du synchrotron de Grenoble (ESRF) [Boc97]. Le dispositif expérimental s'étend sur quelques 200 m^2. Il réunit une variété de détecteurs, des instrumentations de vide, de cryogénie, d'optique, d'électronique et d'acquisition de données.

Si le cœur du système d'acquisition reste réservé à des systèmes électroniques et informatiques spécialisés, le contrôle / commande des instruments conventionnels a quant à lui été réalisé avec un programme LabVIEW. L'ensemble instrumental comporte une trentaine d'éléments : des convertisseurs analogique / numérique, des moteurs pas-à-pas associés aux éléments d'optique et de contrôle du faisceau, un laser de 30 W, des mesures de température et de vide, des photodiodes. Ces instruments sont réunis sur un bus GPIB contrôlé par un PC.

Pendant les acquisitions de données, qui durent plusieurs jours, le programme fonctionne en esclave du programme d'acquisition principal, qui lui transmet des commandes pour modifier les paramètres expérimentaux (présence ou non du faisceau laser, orientation d'un polariseur…). Simultanément, une série d'indicateurs généraux, mesures et signaux d'alarme, est présentée sous forme synoptique à l'opérateur. Enfin, une régulation de position peut être activée pour compenser la dérive du faisceau laser par rapport au faisceau d'électrons du synchrotron.

En dehors des périodes d'acquisition, le dispositif LabVIEW reste actif. En effet une seconde partie du programme permet le contrôle individuel des sous-systèmes, pour le réglage du montage, sa maintenance ou l'amélioration de son fonctionnement. Ainsi, plusieurs développements ponctuels ont permis d'affiner la connaissance et la maîtrise du dispositif.

2.3 L'exemple de PLANCK

L'expérience PLANCK réalisera à partir de 2007 une mesure du rayonnement fossile de l'univers par des détecteurs refroidis à quelques fractions de Kelvin, embarqués sur satellite (ESA) [Tau00]. Le LPSC réalise l'électronique de commande d'un étage cryogénique développé par un laboratoire américain (JPL / Caltech / NASA).

Deux simulateurs locaux des interfaces de cette électronique ont été réalisés sous LabVIEW pour accompagner le travail de développement et de test. Le simulateur de cryogénérateur permet de commander des signaux statiques et dynamiques pour simuler les réponses d'une cinquantaine de capteurs. Le simulateur de satellite quant à lui, communique avec le module électronique via un bus avionique 1553. Il fournit l'interface utilisateur du dispositif, permettant le paramétrage et l'envoi des commandes, ainsi que l'affichage de près de 400 valeurs d'état du système. Il permet aussi l'analyse des réponses des convertisseurs analogique / numérique, et assure les fonctions de sécurité du montage.

3 ATOUTS RELATIFS A L'ENVIRONNEMENT ET AUX METHODES DE LA RECHERCHE

3.1 Spécifications ? Evolution !

L'une des principales fonctions du logiciel de laboratoire consiste à copier un geste expérimental de façon à l'exécuter plus vite, plus longtemps, avec plus de précision, dans un environnement ou à des échelles de taille inaccessibles à l'être humain. De fait, le besoin est souvent exprimé par l'expérimentateur sous la forme de procédures exécutées manuellement ou en imagination. L'utilisation d'un langage *centré sur les traitements* tel que le langage G est donc avantageuse pour la programmation des montages.

La définition des entrées et des sorties de l'automate par l'élaboration d'un *prototype d'interface graphique* (panneau avant du programme), à l'aide d'une variété d'objets graphiques imitant les éléments de face avant habituels des instruments, peut compléter facilement ce cahier des charges initial.

Cependant, l'utilisation d'un montage expérimental suppose des tâtonnements et des évolutions. Si précises qu'elles soient, les spécifications initiales ne sont pas faites pour durer. Ainsi l'implémentation dans une procédure séquentielle trop linéaire du fonctionnement décrit par le demandeur au début du projet étouffe ou ralentit très vite le processus de développement. La structure de la *machine à états finis*, particulièrement lisible sous LabVIEW, répond très efficacement à ce problème. En décomposant le fonctionnement demandé en procédures ou actions élémentaires, asservies chacune à un événement (pression d'un bouton, niveau d'un capteur, condition logique) et réunies dans une telle structure, on rend le code plus lisible tout en laissant à l'utilisateur le soin d'orchestrer lui-même l'enchaînement des actions.

Il est ensuite facile de définir de nouvelles actions, de les regrouper ou de les décomposer au gré des progrès réalisés dans la définition du fonctionnement voulu.

3.2 L'instrument virtuel : une réalité pour l'expérimentateur

Il existe une forte analogie entre l'instrument matériel et le composant fonctionnel abstrait manipulé dans la programmation classique : tous deux sont successivement paramétrés (configurés) puis exécutés (déclenchés) afin de renvoyer un résultat (mesure, nouvel état). L'organisation des composants fonctionnels d'un programme suivant une arborescence hiérarchique (d'ailleurs générée automatiquement sous LabVIEW) permet de maintenir la clarté du code à tous les niveaux de lecture. Dans la pratique expérimentale, on constate que les ensembles instrumentaux s'organisent suivant le même schéma. Sous LabVIEW, la composition d'une collection instruments (à GRAAL par exemple) forme un nouvel instrument, donnant toute sa signification à la notion de montage.

En maintenant cette cohérence modulaire entre le programme et les instruments, il devient facile d'exploiter de nouvelles combinaisons instrumentales. Ainsi, à GRAAL, en partant uniquement des instruments en place, on a pu développer un dispositif de positionnement de lames d'ondes, qui a lui-même fourni les bases d'un polarimètre évolué pour la mesure des paramètres de Stokes du faisceau laser.

3.3 Etudiants et chercheurs s'impliquent rapidement

Les équipes de travail des laboratoires sont pratiquement toujours composées d'un

groupe technique et de quelques physiciens et étudiants. Le passage est donc rapide de la culture scientifique à la culture technique. Les étudiants et stagiaires doivent exploiter des montages automatisés dès leur arrivée alors qu'ils n'ont, au mieux, qu'une connaissance académique de la programmation scientifique.

La lisibilité de l'interface graphique associée à tous les modules de LabVIEW et la syntaxe conviviale du langage graphique contribuent à faciliter l'implication de tous les intervenants. La possibilité d'inclure des formules de calcul et des structures de contrôle dans un formalisme proche du langage C permet quant à elle un passage immédiat de la feuille de calcul du chercheur au programme d'analyse des données.

4 ATOUTS TECHNIQUES DIRECTS

Le développement des langages de programmation les plus connus s'est largement concentré sur la maîtrise du calcul, du traitement et de la modélisation des données. Ces progrès ne répondent que très partiellement aux problématiques techniques des laboratoires, qui doivent communiquer avec des automates, maîtriser des processus physiques dynamiques, réagir aux sollicitations d'un environnement non déterministe [Lee02].

4.1 Des instruments qui parlent…

La modularité fonctionnelle des langages a fourni deux apports majeurs : l'élimination de la redondance des traitements et la clarification de leur sémantique d'une part, la possibilité de constituer des bibliothèques standard d'autre part. Dans le domaine spécifique de l'expérimentation, disposer de bibliothèques dédiées à la commande et à la communication avec les appareils ne s'impose pourtant pas à l'esprit de nombreux responsables de projets. L'option économique procurée par la réutilisation d'outils existant dans un laboratoire aboutit facilement à l'usage d'un langages "orienté calcul" (Fortran…) ou dépourvus de primitives instrumentales de haut niveau (environnements C usuels).

Au mieux, ce choix implique une perte de temps importante dans la réécriture et le test des procédures de haut niveau indispensables pour ramener la jungle des matériels et des protocoles à la simplicité conceptuelle d'un échange de données. La programmation du port série RS232, sans doutes intéressante dans le cadre d'un cours de C, a mobilisé des légions de techniciens.

La popularisation de LabVIEW a été prise en compte par la plupart des constructeurs d'instruments qui mettent à disposition des primitives de très haut niveau. Le langage G a du vocabulaire. Le développement se concentre donc directement sur les problèmes techniques de haut niveau du projet. Dans un laboratoire, cet aspect a une répercussion directe sur le cycle résultats / amélioration de l'expérience décrit plus haut.

4.2 Un programme qui répond !

Depuis une quinzaine d'années, la multiplication des applications embarquées (moyens de transport, dispositifs ménagers…) et distribuées (applications des réseaux informatiques) a mis en évidence un certain nombre de lacunes conceptuelles des langages et des architectures issues des travaux de Von Neumann, en particulier pour la gestion de la temporalité. Aujourd'hui des langages comme le G ont su évoluer pour intégrer les premières formes de réponses à ces problématiques : exécution concurrente (pseudo - parallélisme), synchronisation de processus, temps réel.

En effet le modèle d'exécution par *flots de données* sur lequel repose LabVIEW constitue l'un des plus pertinents pour la

programmation d'applications réactives. L'association d'un design multiprocessus et d'une organisation en machine à états finis est particulièrement féconde [Lee02].

La syntaxe graphique du langage G confirme cette prédisposition. Elle fournit naturellement une vue du graphe d'activation des processus [ZB99] du programme. L'écriture des processus itératifs parallèles sous la forme de blocs "while" placés côte à côte est d'une remarquable lisibilité. Tout comme la modularité fonctionnelle, cette modularité des processus prend à l'écran la forme intuitive de boîtes de code.

Rapprochée du concept d'instrument virtuel, l'approche multiprocessus et les outils de communications qui l'accompagnent permettent de réaliser de véritables *instruments virtuels programmables*. Le mécanisme de *queue* implémente un bus de communication virtuel. Le mécanisme de *notification* implémente un déclenchement synchrone (trigger) virtuel.

Nous avons pu mesurer la puissance de cette possibilité sur PLANCK. L'application du simulateur de satellite (qui constitue l'interface de contrôle/commande de l'appareil) a été conçue suivant une architecture fonctionnelle typique des systèmes temps réel en quatre niveaux [ZB99] :

- Un niveau de contrôle direct, assurant les opérations de communication avec le matériel ;
- Un niveau de collecte et d'archivage, assurant *à sa fréquence propre* la mise à jour d'une base de données sur l'état du système ;
- Un niveau de gestion de l'information, exploitant *à sa fréquence propre* les informations fournies pour l'affichage et la surveillance automatique des informations;
- Un niveau d'interfaçage homme-machine.

Le découplage des opérations d'acquisition et d'exploitation des mesures permet une excellente adaptation à des contraintes temporelles différentes : un seul PC gère aisément et rapidement toutes les tâches, incluant l'émission d'un signal électrique de sécurité toutes les secondes et un moteur de surveillance et d'alarmes à trois degrés de criticité. La lisibilité du programme (donc l'évolutivité, la maintenance, la fiabilité) est aussi améliorée. Ainsi le *découplage temporel* des processus procure des avantages aussi immédiats que ceux du découpage fonctionnel des traitements.

La gestion des données au travers d'une forme de base de données globale a rendu facile l'affichage non pas d'un moniteur synoptique, mais de 5 (autant qu'on veut !) écrans traduisant efficacement différents niveaux de lecture du montage.

4.3 La permanence des moyens de réglage et de validation

Des mécanismes comme le typage fort (supporté par LabVIEW sous le nom de "type strict") ont permis d'améliorer grandement la détection statique (avant l'exécution) d'erreurs liées au traitement des données. En ce qui concerne la temporalité, de tels mécanismes n'en sont qu'à leurs balbutiements, alors que c'est là le point critique des applications réactives.

LabVIEW procure les moyens d'améliorer cette lacune (commune à tous les environnements de développement actuels), par une utilisation méthodique de la puissance interactive des panneaux avant des modules pour *valider* leur fonctionnement pendant et après le développement. En effet, si une partie

seulement des modules fonctionnels est destinée à l'affichage, tous peuvent être conçus comme des "instruments" interactifs à part entière, intégrant non seulement des entrées-sorties destinées aux autres modules du programme, mais aussi des terminaux de réglage, de test et de validation. Ainsi sur GRAAL, les modules de correction PID récursifs contiennent des graphes pour le réglage des paramètres de gain, et la vérification de leur comportement.

En intégrant les moyens de validation dans le code, on améliore grandement sa fiabilité et sa maintenance, en particulier sur ce qui touche à la communication avec les instruments.

5 CONCLUSION

Les réalisations du Service Détecteurs et Instrumentation du LPSC sur les projets GRAAL et PLANCK, mettent en évidence une forte concordance entre les spécificités de LabVIEW et les contraintes de la recherche. En intégrant dans un environnement convivial les progrès récents réalisés dans le domaine des applications réactives, LabVIEW contribue à populariser des concepts qui devraient lui permettre de consolider son rôle au sein du laboratoire.

6 BIBLIOGRAPHIE

[Boc97] J.P. Bocquet et al., « GRAAL : a polarized gamma-ray beam at ESRF », Second ELFE workshop on hadronic physics, Saint-Malo, 23-27 septembre, 1997, Nucl. Phys. A622 (1997) 124c

[Lee02] Edward A. Lee, « Embedded Software », Advances in Computers, Marvin V. Zelkowitz, 56, 2002.

[Tau00] J. A. Tauber, « The Planck Mission », New Cosmological Data and the Values of the Fundamental Parameters, International Astronomical Union. Symposium no. 201. Manchester, Angleterre, Août 2000

[ZB99] L. Zaffalon, P. Breguet, « Programmation concurrente et temps réel en ADA 95 », Presses polytechniques et universitaires romandes, 1999.