Développement d'un framework en Java pour le contrôle-commande de la caméra du télescope LSST

Bernard Amade, Eric Aubourg, Etienne Marin-Matholaz, Françoise Virieux
IN2P3/CNRS, Astroparticule et Cosmologie (APC), Paris

LSST : *Large Synoptic Survey Telescope*

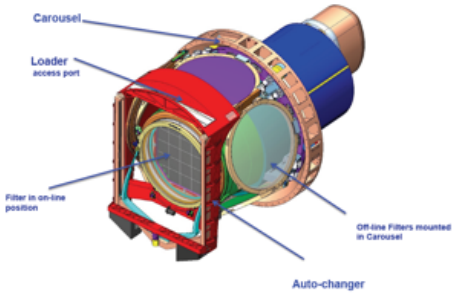
Grand télescope au sol (première lumière en 2019), caméra de 3.2 Giga pixels. Conçu pour obtenir des images profondes de la moitié du ciel tous les 4 jours. Projet des agences USA DOE (construction de la caméra) et NSF (télescope et réduction des donnés) et de l'IN2P3 (APC,CPPM,CCIN2P3,LAL,LMA,LPC-Clermont,LPNHE,LPSC, ≈ 10% de la construction de la caméra).

Principales contributions de l'IN2P3

- en informatique : le développement du logiciel de contrôle-commande de l'ensemble de la caméra et le logiciel de contrôle-commande du changeur de filtres,
- en mécanique la conception et l'assemblage du changeur de filtres de la caméra
- en électronique : tests de réception de 25% des CCD livrés (225 CCD de 4k×4k), développement, qualification, production et tests des circuits de lecture (ASPIC) et de cadencement (CABAC) des CCD, le micro-code FPGA de contrôle des CCD,

Changeur de filtres de la caméra de LSST

6 filtres sont disposés autour de la caméra, le changeur de filtres permet de mettre devant le plan focal le filtre souhaité.
diamètre d'un filtre : ≈ 60cm
poids d'un filtre : entre 40 et 60kg



La caméra de LSST

La caméra est composée d'une quinzaine de sous-systèmes, parmi ceux-ci :

- l'échangeur de filtres
- la gestion de la température
- la gestion du froid
- la gestion de la puissance électrique
- la gestion du vide
- l'obturateur
- l'acquisition des données

Architecture du logiciel

Le CCS, "Camera control system", a pour objectif de contrôler et coordonner les actions des différents sous-systèmes de la caméra. Il est l'interface entre le logiciel de contrôle du télescope et les sous-systèmes de la caméra.

Un canevas pour créer des sous-systèmes Le CCS fournit des classes Java permettant de construire rapidement un sous-système en assemblant des modules réutilisables. Il fournit aussi des outils : consoles, écrans de contrôle, bases de données, bus de communication.

Communication La communication entre les différents sous-systèmes repose sur trois bus de messages sur Ethernet :

- *Command Bus* : achemine les commandes vers les différents sous-systèmes
- *Status Bus* : transmet à l'observatoire les informations de configuration et d'état des sous-systèmes
- *Log Bus* : recueille les données de chaque sous-système à des fins de détection d'erreur et de diagnostique.

Un logiciel fait pour durer 30 ans Entre l' écriture du premier prototype en 2006 et la fin de vie du télescope il se sera écoulé 30 ans. La **durabilité** et la **maintenabilité** du logiciel sont donc des paramètres importants. C'est pourquoi nous utilisons dès que possible des outils ou librairies existantes, en veillant à être le moins possible dépendant de ces outils en utilisant des Interfaces Java. Exemple d'outils : JGroups ou JMS pour la couche de communication, Hibernate pour l'interface avec les bases de données, Groovy pour le langage de script et la description d'un sous-système. Beaucoup de nos efforts portent aussi sur **les tests et la validation** du logiciel. Nous fournissons aux développeurs de sous-systèmes des outils de tests à faire réaliser sur notre serveur d'intégration continue Jenkins.

Pourquoi développer un logiciel spécifique ?

Nous avons envisagé PVSS et EPICS mais nous les avons écartés :

- courbe de prise en main exponentielle
- documentation insuffisante
- adapter ces logiciels à nos besoins requiert souvent l'écriture de beaucoup de code de haut et de bas niveau
- incertitude sur la durée de vie de ces logiciels

Pourquoi Java ?

- pourquoi pas java ?
- un langage objet est bien adapté à la description des objets physiques que nous manipulons
- la modularité permet un développement reparti entre plusieurs équipes
- le nombre et la richesse des librairies disponibles en Java sont énormes
- communiquer avec les pilotes du matériel est facilité par JNI
- bonne compétence de Java dans l'équipe
- réutilisation de code écrit par le SLAC pour le projet Fermi autour de JAS (consoles, analyse des données télémétriques)

Pour en savoir plus

Large Synoptic Survey Telescope

Au laboratoire APC de l'IN2P3, Eric Aubourg, Bernard Amade, Etienne Marin-Matholaz et Françoise Virieux conçoivent et développent le CCS et le logiciel de contrôle de l'échangeur de filtres (FCS) en collaboration avec le laboratoire SLAC, Stanford University, en Californie.
mailto : emarin@apc.univ-paris7.fr, virieux@in2p3.fr, bamade@in2p3.fr

Lire les numéros 16 et 17 de **La Lettre Informatique de l'IN2P3**
Le site français du projet LSST : http ://lsst.in2p3.fr
Le site américain du projet LSST : http ://www.lsst.org/lsst/