



JMMC-MIN-2000-0001

Date : 20/10/2006

# JMMC

## MINUTES DE LA REUNION TECHNIQUE LOGICIEL

*Lyon, 17 Octobre, 2006*

### Participants:

*Isabelle Tallon-Bosc  
Alain Chelli  
Renaud Foy*

*Michel Tallon  
Gilles Duvert*

*Eric Thiebaut  
Gérard Zins*

### Ordre du jour

- 1 Objectifs de la réunion**
- 2 Descriptif de la démarche du groupe de Lyon**
  - 2.1 Pourquoi Yoga / Litpro
  - 2.2 Caractéristiques (démonstration Goutelas)
  - 2.3 Ce qu'il reste à faire
- 3 Descriptif de ce qui est développé par le centre de réalisation**
  - 3.1 Etat d'avancement
  - 3.2 Ce qu'il reste à faire pour une livraison version publique
- 4 Conclusions**
  - 4.1 Compatibilité des 2 démarches ?
  - 4.2 Ou travaux en parallèle comme décidé lors du dernier CS ?
  - 4.3 Responsabilité du groupe Model Fitting : qui et pour faire quoi?

### **1 Objectifs de la réunion**

L'objectif de la réunion est de faire le point d'avancement sur les travaux menés au sein du groupe de recherche à Lyon (GL) et ceux menés par le centre de réalisation à Grenoble (CRG), et de définir d'un commun accord des actions à mener, en prenant en compte le retour d'expérience des essais effectués sur des données réelles avec des utilisateurs.

## 2 Descriptif de la démarche du groupe de Lyon

### 2.1 Pourquoi Yoga / Litpro

Un premier prototype, baptisé LIT, fut fonctionnel à l'automne dernier; écrit en Yorick, il a servi de base pour le développement, sous environnement MCS, d'une version publique (cf. Section 3.1).

Suite d'une part aux premiers essais effectués sur des données réelles, en particulier sur des données AMBER avec Romain Petrov et Florentin Millour, et d'autre part aux réflexions menées avec Nicolas Nardetto, Denis Mourard et Merième Chadid au sujet de l'intégration des modèles chromatiques, il s'est avéré utile, voire indispensable, de restructurer le logiciel pour offrir davantage de flexibilité. D'où la naissance de LITpro en juin dernier, qui devrait permettre d'atteindre plus facilement les objectifs fixés répertoriés dans le cahier des charges initial (document JMMC-SPE-2300-0001). LITpro est englobé dans Yoga, un environnement logiciel de développement d'algorithmes incluant MIRA, logiciel de reconstruction d'image et des modules communs tels que Yorick et Yeti pour le langage, OptimPack pour l'optimisation, OILib pour la gestion des fichiers OIFits.

Contrairement à LIT, LITpro n'a pas de gestion dynamique des briques objets et le modèle n'est pas une somme pondérée de modèles élémentaires. Avec LITpro, la pondération est dans les modèles élémentaires mêmes, effective ou non selon le désir de l'utilisateur. Ce dernier ne peut plus bâtir son modèle au fur et à mesure, de façon dynamique. Par contre, il lui suffit pour ajuster ses données de charger un fichier de configuration. Celui-ci définit les 'targets' à ajuster, une 'target' étant principalement caractérisée par :

- l'identifiant d'un objet,
- le ou les fichiers contenant les données d'observation pour cet objet,
- la définition du modèle,
- les paramètres à ajuster avec les contraintes associées,
- la sélection des types de données à ajuster,
- la sélection du  $\chi^2$  à utiliser pour la minimisation, pour chaque type de données.

Le fichier de configuration chargé, il ne reste plus à l'utilisateur qu'à lancer la procédure d'ajustement puis à visualiser les paramètres de sortie, ainsi que les caractéristiques du fit (nombre d'itérations, matrices de covariance et corrélations, etc).

### 2.2 Caractéristiques (démonstration Goutelas)

Ces caractéristiques ont été brièvement montrées à l'aide du mode « Practice » du logiciel, mis au point pour la séance de Travaux Pratiques de l'école « Observation and data reduction with the VLTI ». A Goutelas a en effet été présenté le logiciel Yoga (sans MIRA) enrichi de fonctions 'haut niveau' pour en faciliter l'utilisation, tout en laissant accessibles aux utilisateurs avertis les fonctions avancées.

Nous pouvons lister ici les fonctionnalités existantes:

- bibliothèque de modèles élémentaires, identique à celle de LIT avec des lois d'assombrissement centre-bord supplémentaires,
- possibilité d'ajuster plusieurs targets en même temps (par ex. objet et

calibrateur), les données pouvant être lues dans plusieurs fichiers pour chaque target,

- calcul des gradients par différences finies,
- bornage des paramètres possible,
- fixation des paramètres possible,
- calcul des matrices de covariance, de corrélation et des barres d'erreur,
- optimisation: seuls les résidus nécessaires sont calculés,
- visualisations (plot des modèles, de leur spectre, de la couverture uv des données, etc)

### 2.3 **Ce qu'il reste à faire**

Ce qu'il reste à faire peut être divisé en 2 groupes de tâches. Les premières (T1) sont nécessaires pour rapidement rendre utilisable le logiciel par des membres du groupe Model Fitting. Les secondes (T2) relèvent typiquement de la R&D et doivent être menées sur un plus long terme pour rendre Yoga pleinement opérationnel, la longueur du terme dépendant des forces humaines en R&D dont nous disposerons au sein du groupe.

Objectifs des T1 :

- documentation, si possible dynamique,
- lecture du fichier de configuration écrit par l'utilisateur (« parser »),
- écriture d'un fichier de configuration (pour sauvegarde et modification),
- affichage d'un rapport sur les données lues,
- implantation d'un premier modèle chromatique.

Objectifs des T2 :

- optimisation globale : bibliothèque de différents moteurs d'ajustement (ex. ASA, pseudo clean,...) --> choix du fitter par l'utilisateur
- outils pour rendre efficace la recherche des minima locaux ( $\chi^2$  mapper,  $\chi^2$  slicer..)
- implantation d'un modèle instrumental,
- implantation d'un modèle dit « astrophysique »
- enrichissement des visuels graphiques, généré par les besoins.

Entre T1 et T2, figurerait l'action « écriture d'un simulateur de données OIFits », outil indispensable pour pouvoir disposer de données simulées et réaliser les tâches T2, mais dont l'absence n'empêche pas le début d'exploitation de Yoga.

## **3 Descriptif de ce qui est développé par le centre de réalisation**

### 3.1 **Etat d'avancement**

Une première maquette du logiciel a été développée par Ludovic Saugé, et testée à partir de juin. Les résultats obtenus sont comparables à ceux du logiciel Yorick qui a servi de référence pour les développements. Cette maquette ne comprend pas d'interface graphique, et est interrogeable via des commandes transitant via le service de messagerie de l'environnement commun MCS. Il bénéficie d'ailleurs de tous les autres services de cet environnement tel que le système de log ou de la gestion des erreurs.

Ludovic a également travaillé sur l'implémentation d'un algorithme génétique comme l'une des solutions envisageables pour le problème des minima locaux. Les résultats obtenus sont plutôt encourageants.

Après le départ de Ludovic en août, les développements ont été repris par Laurence Gluck et Guillaume Mella. Laurence travaille sur la partie serveur, correspondant au logiciel Yorick, tandis que Guillaume est en charge de l'interface graphique. La première étape a consisté à consolider la structure du programme (diagramme de classes et diagramme de séquence) et définir les interfaces entre le serveur et l'interface graphique. Cette étape est plus ou moins terminée. Les solutions techniques utilisées sont celles définies/recommandées par MCS, à savoir C/C++ pour le serveur et Java pour l'interface graphique, avec une communication basée sur TCP/IP respectant le protocole MCS. Une maquette de l'interface graphique est actuellement opérationnelle, dont le but est de montrer les possibilités d'affichage et de présentation des données/résultats de l'ajustement ; points particulièrement délicats dans le cadre de ce développement.

### **3.2 Ce qu'il reste à faire pour une livraison version publique**

Les développements restant à effectuer sont :

- Finalisation des développements du serveur (délai 2-3 mois),
- Validation du logiciel (délai 1 mois)
- Spécification, à partir de la maquette, de l'interface graphique (délai 1 mois, avec itération avec le groupe représentatif des utilisateurs)
- Développement de l'interface graphique (délai 5-6 mois),
- Rédaction du manuel utilisateur

L'objectif de produire une version publique pour le printemps 2007.

## **4 Conclusions**

### **4.1 Compatibilité des 2 démarches ?**

Les développements ne sont pas incompatibles. Toutefois, il est souligné que les premiers essais effectués sur des données réelles ont conduit à des évolutions significatives du logiciel Yorick et à la naissance de Yoga. De telles évolutions devront être considérées pour la version en cours de développement à Grenoble, et par conséquent, la transcription du logiciel en C/C++ est sûrement prématurée.

Sur ce constat, une nouvelle voie pour le développement et la distribution du logiciel d'ajustement de modèles, pour la version publique, est envisagée. Cette nouvelle démarche consisterait à :

- Utiliser Yoga (version Yorick du logiciel) comme serveur,
- Coupler l'interface graphique Java à ce logiciel, en offrant un accès restreint au logiciel Yoga; seules les fonctionnalités valides et robustes sont disponibles via cette interface.

Les intérêts d'une telle solution (déjà envisagée par Eric), sont :

- De proposer un logiciel au grand public, bénéficiant des dernières avancées en terme de R&D.

- D'attendre la validation du processus d'ajustement de modèles par la confrontation avec des données réelles, tout en distribuant un logiciel disposant d'un minimum de fonctionnalités pour présenter un intérêt.

Les conséquences directes de cette solution, sont :

- L'arrêt des développements de la version C/C++ ; la structure ayant été mise en place pouvant toutefois servir à une version future du logiciel lorsque la R&D sera terminée,
- La mise à disposition de personnel technique du CRG pour le développement de l'interface graphique.

En plus du support pour le développement de l'interface graphique, le CRG pourra assurer la gestion de configuration, la distribution du logiciel et le suivi des anomalies. Le développement et la maintenance du logiciel Yorick restent quant à eux sous l'entière responsabilité du GL ; aucune des méthodes/règles de développements définies par le CRG n'est applicable à ces développements.

Il est convenu que chaque partie réfléchisse à cette solution, et qu'une nouvelle discussion ait lieu au plus tôt (d'ici 1 à 2 semaines) pour statuer.

#### **4.2 Ou travaux en parallèle comme décide lors du dernier CS ?**

Voir ci-dessus.

#### **4.3 Responsabilité du groupe Model Fitting : qui et pour faire quoi?**

Ce sujet n'a pu être abordé lors de cette réunion.