

	<p>JMMC-SPE-2210-0001</p> <p>Version : 2.0</p> <p>Date : 15/09/2011</p>
---	---

JMMC

JMCS - CAHIER DES CHARGES

Sylvain LAFRASSE (sylvain.lafrasse@obs.ujf-grenoble.fr)
JMMC

<p>Author : Sylvain LAFRASSE</p> <p>Institute : JMMC</p>	<p>Signature :</p> <p>Date :</p>
<p>Approved by : Guillaume MELLA</p> <p>Institute : JMMC</p>	<p>Signature :</p> <p>Date :</p>
<p>Released by : Gilles DUVERT</p> <p>Institute : JMMC</p>	<p>Signature :</p> <p>Date :</p>

TABLE OF CONTENTS

1	INTRODUCTION	4
1.1	Objectif	4
2	DESCRIPTION GÉNÉRALE	4
2.1	Contexte	4
2.2	Utilisateurs	4
2.3	Aperçu des besoins fonctionnels	4
2.4	Environnement de développement	5
2.5	Documentation	5
3	BESOINS	5
3.1	Uniformisation de la fenêtre d'à-propos/écran de démarrage	5
3.2	Uniformisation de la gestion des retours d'erreurs et commentaires	6
3.3	Uniformisation de la gestion des journaux d'exécution	6
3.4	Uniformisation des menus et raccourcis claviers	6
3.5	Uniformisation de la génération/visualisation de l'aide utilisateur	6
3.6	Uniformisation de la gestion des préférences	7
3.7	Uniformisation de la gestion d'ouverture/sauvegarde des fichiers	7
3.8	Uniformisation de l'interface « ligne de commande »	7

1 INTRODUCTION

Le but de ce document est de recenser les fonctionnalités que JMCS doit fournir.

1.1 Objectif

L'objectif de JMCS est de réunir au sein d'un même module logiciel facilement réutilisable, un maximum de fonctionnalités partagées par toutes les applications Java graphiques du JMMC.

JMCS nous permettra d'uniformiser les interfaces et l'utilisation de nos logiciels, en offrant une intégration aussi poussée que possible aux principaux systèmes d'exploitations du marché.

Par cette démarche, nous souhaitons augmenter le niveau de finition de nos interfaces graphiques et donc la satisfaction de nos utilisateurs, tout en facilitant la maintenance corrective et évolutive de nos logiciels.

2 DESCRIPTION GÉNÉRALE

2.1 Contexte

Dans le cadre du développement des logiciels Java du JMMC SearchCal et ModelFitting, nous avons été amenés à factoriser une partie du code pouvant être réutilisé de part et d'autre. Nous avons donc créé le module JMCS pour contenir tous ces développements. Le but est d'enrichir ce module, et de le peaufiner encore.

2.2 Utilisateurs

JMCS est destiné à deux classes d'utilisateurs.

La première est constituée des développeurs du JMMC, qui sont amenés à utiliser JMCS pour concevoir nos logiciels graphiques Java. Pour eux, JMCS doit être simple et rapide de mise en œuvre.

Mais à travers les produits finis que les développeurs créent avec JMCS, une deuxième classe d'utilisateurs est aussi directement concernée: les scientifiques. Ce sont les utilisateurs finaux de nos logiciels, qui bénéficieront de cette architecture pour simplifier leur apprentissage et leur compréhension de nos programmes. Ils sont représentés au JMMC par des groupes de travail associés à chacun des logiciels que nous produisons.

2.3 Aperçu des besoins fonctionnels

Les objectifs principaux du développement du module JMCS sont les suivants :

- Offrir aux scientifiques l'uniformité de nos interfaces graphiques, et leur intégration aux systèmes d'exploitation sous-jacents;
- Offrir aux développeurs un maximum de fonctionnalités « gratuites » et simples de mise en œuvre.

Les thématiques à couvrir sont:

- Uniformisation de la fenêtre d'à-propos/écran de démarrage ;

- Uniformisation de la gestion des retours d'erreurs et commentaires ;
- Uniformisation de la gestion des journaux d'exécution ;
- Uniformisation des menus et raccourcis claviers ;
- Uniformisation de la génération/visualisation de l'aide utilisateur ;
- Uniformisation de la gestion des préférences ;
- Uniformisation de la gestion d'ouverture/sauvegarde des fichiers ;
- Uniformisation de l'interface « ligne de commande ».

2.4 Environnement de développement

JMCS repose sur Java en version 1.5 ou ultérieure, et doit offrir ses fonctionnalités aux applications JMMC tant Java WebStart qu'autonomes (JARs).

L'environnement technique de développement est MCS.

2.5 Documentation

Comme JMCS est destiné à deux classes distinctes d'utilisateurs, il doit offrir deux documentations différentes, toutes deux en anglais :

- Une documentation à destination des développeurs devant utiliser JMCS, entre autres composée des commentaires du code (dans le source même de JMCS) servant à la génération automatique d'un guide des APIs, ainsi qu'un guide technique d'utilisation de JMCS avec exemples pratiques de mise en œuvre.
- Une documentation à destination des scientifiques utilisant nos logiciels basés sur JMCS, qui pourra être intégrée par défaut dans le système de gestion de la documentation devant être fourni par JMCS.

3 BESOINS

Les besoins exprimés ci-après le sont dans l'ordre de priorité de réalisation.

3.1 Uniformisation de la fenêtre d'à-propos/écran de démarrage

Le but est de proposer un mécanisme standardisé d'affichage des informations concernant nos applications. Les informations à afficher sont les suivantes :

- Le logo JMMC et lien vers la page d'accueil du site web du JMMC ;
- Lien vers la page web de présentation générale du logiciel ;
- Nom et numéro de version (avec mécanisme de systématisation de leur gestion) ;
- Liste des dépendances et modules ;
- Licence et copyright ;
- Texte libre pour d'éventuelles informations non recensées ;
- Date de compilation et version du compilateur utilisé.

Ces informations doivent toutes être affichées dans la fenêtre d'à-propos de l'application, et un sous-ensemble (logo JMMC, nom et numéro de version de l'application) pourra être aussi affiché dans la fenêtre de chargement du logiciel.

3.2 Uniformisation de la gestion des retours d'erreurs et commentaires

Le retour de commentaires/erreurs par l'utilisateur se fera dans une fenêtre vouée à recueillir ses remarques. Seront adjointes automatiquement au rapport transmis les informations suivantes :

- La version du logiciel, et la version de la JVM utilisée ;
- Le journal d'exécution ;
- Les propriétés du système hôte ;
- Les préférences du logiciel, et celles partagées ;
- L'adresse mail de l'utilisateur pour contact ultérieur (optionnel) ;
- Des éventuelles données spécifiques à l'application (par exemple la requête SearchCal pour pouvoir reproduire le problème, éventuellement le résultat).

Le mécanisme de retour de l'information reste à définir.

3.3 Uniformisation de la gestion des journaux d'exécution

La gestion des journaux d'exécution est confiée au mécanisme standard Java Logger.

L'intégration systématique d'un outil du style « InforMatrix LogGui » assurera aux développeurs un débogage facilité.

3.4 Uniformisation des menus et raccourcis claviers

Le but ici est de présenter aux utilisateurs, en plus des menus spécifiques à nos applications, une arborescence de menus commune à toutes nos applications pour les fonctions standard, comme :

- Menu Fichier : ouverture, sauvegarde, impression, ... ;
- Menu Edition : copier, couper, coller, ... ;
- Menu Application : quitter, préférences, à-propos, aide utilisateur, retour et commentaires ;
- Tout autre élément factorisable.

Les standards d'arborescence des menus et de leurs raccourcis claviers correspondant à toutes ces fonctions communes sont à recenser sur chacune des grandes plates-formes d'exécution (Mac OS X, Linux, Windows), et à utiliser en conséquence pour les menus communs identifiés.

Tout cela doit se faire :

- Dans l'optique d'une intégration aussi poussée que possible au système d'exploitation faisant fonctionner l'application, afin de ne pas nécessiter de l'utilisateur une adaptation superflue à nos logiciels ;
- De manière dynamique, pour ne pas avoir à distribuer différentes versions de nos logiciels suivant la plate-forme à laquelle ils sont destinés.

3.5 Uniformisation de la génération/visualisation de l'aide utilisateur

Le but est de proposer un dispositif de présentation de la documentation utilisateur intégré à toutes nos applications.

Pour chacun de nos logiciels, un scientifique du groupe de travail associé est désigné pour rédiger la documentation utilisateur. Il reste donc à définir avec les scientifiques un format de données approprié.

La documentation doit être si possible indexée, avec possibilité d'ancrage pour références contextuelles de part et d'autres de nos interfaces graphiques (boutons « help »).

Par exemple, la gestion de l'aide faite par le logiciel TOPCAT offre une bonne illustration du système désiré.

3.6 Uniformisation de la gestion des préférences

Les objectifs à atteindre ici sont :

- Offrir un mécanisme uniforme de stockage et d'accès aux préférences;
- Stocker le fichier de préférences dans l'endroit consacré du système hôte ;
- Proposer une fenêtre de préférence par défaut, adaptable à chaque application ;
- Détecter et recréer un fichier de préférence corrompu ou manquant ;
- Détecter et prendre en compte un changement de version de l'application ;
- Avoir des préférences communes à plusieurs de nos applications ;
- Mettre à disposition des développeurs des préférences « cachées », pour définir par exemple le niveau de log et tout autre option spécifique au logiciel concerné sans avoir à le recompiler.

Le format des fichiers de préférence reste à définir.

3.7 Uniformisation de la gestion d'ouverture/sauvegarde des fichiers

Il faut :

- Proposer systématiquement l'enregistrement des documents non encore sauvegardés lors de la fermeture de nos applications ;
- Afficher autant que possible le navigateur de fichier par défaut du système d'exploitation hôte.

3.8 Uniformisation de l'interface « ligne de commande »

Il faut prévoir l'intégration de mécanismes permettant des interactions avec nos applications lors de leur lancement en ligne de commande.

De plus, les actions suivantes doivent être standardisées dans toutes nos applications:

- Aide utilisateur sur les options utilisables en ligne de commande;
- Affichage du numéro de version ;
- Configuration des sorties de journaux d'exécution.