



IFT3051 - E05 - UdeM

Ptidej UI - Mehdi Lahlou

<http://www.iro.umontreal.ca/~lahloume>



- Premier contact de l'utilisateur avec une application [Attrait + langue acquise]
- Prise en main [Conformité]
(Expérience, Intuition développée)
- Accès aux fonctionnalités (Rapidité)
[Groupés + Différentes sources]
- Personnalisation
[Espace de travail] (Goût)



Ptidej UI v1.0:

- + Application innovante
 - + Fonctionnalités intéressantes
 - Interface peu intuitive
 - Interface unilingue
 - Espace de travail rudimentaire
- Délaissée rapidement
- Durée de vie ... courte

Ptidej UI v2.0:

+ Application innovante

+ Fonctionnalités intéressantes

Mettre en valeur

+ Interface intuitive

+ Principes d'usabilité

+ Interface multilingue

+ Espace de travail complet

→ Faciliter la tâche à effectuer

→ Durée de vie ... longue

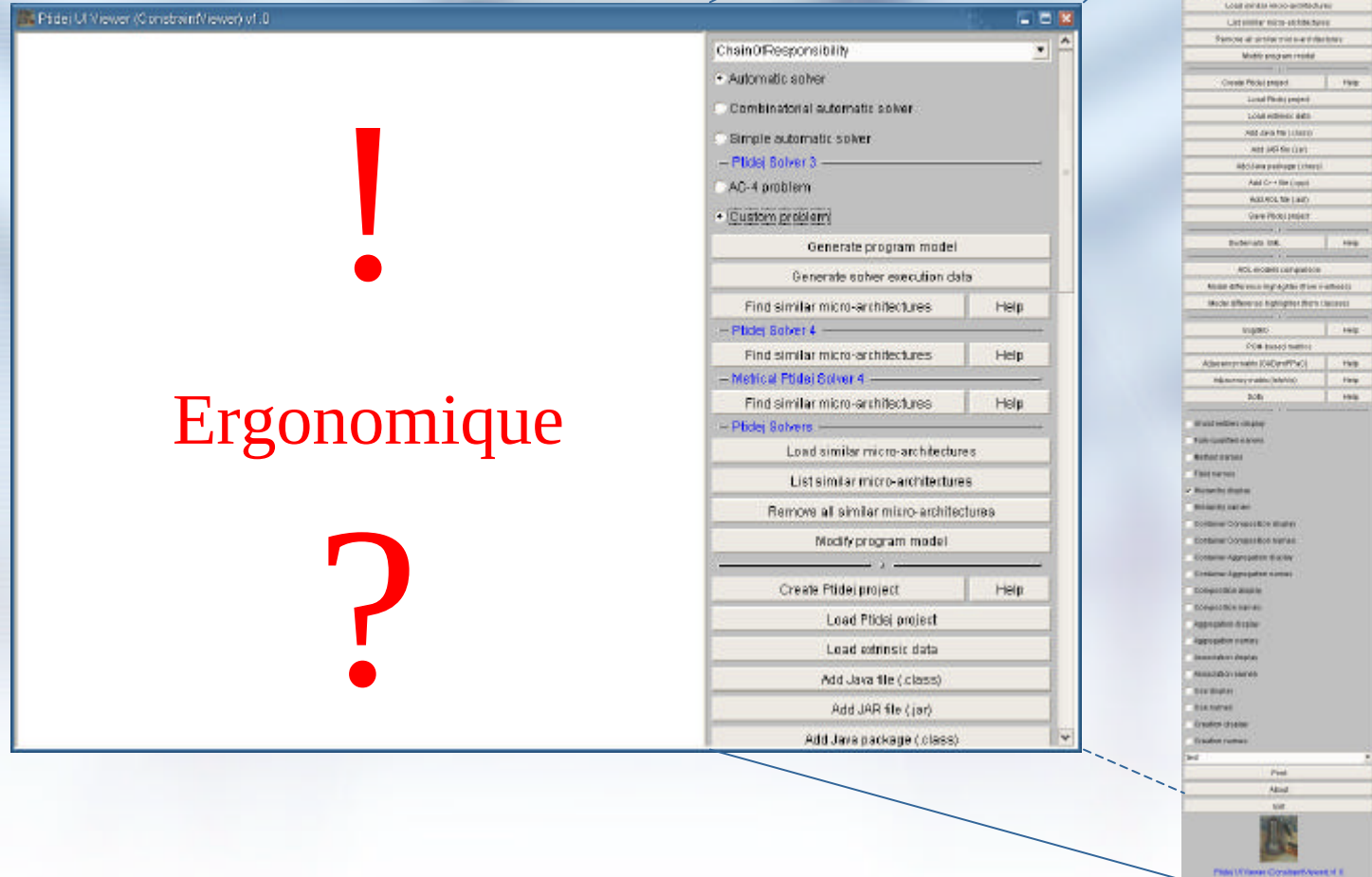


Comparaison - AVANT (v1.0)

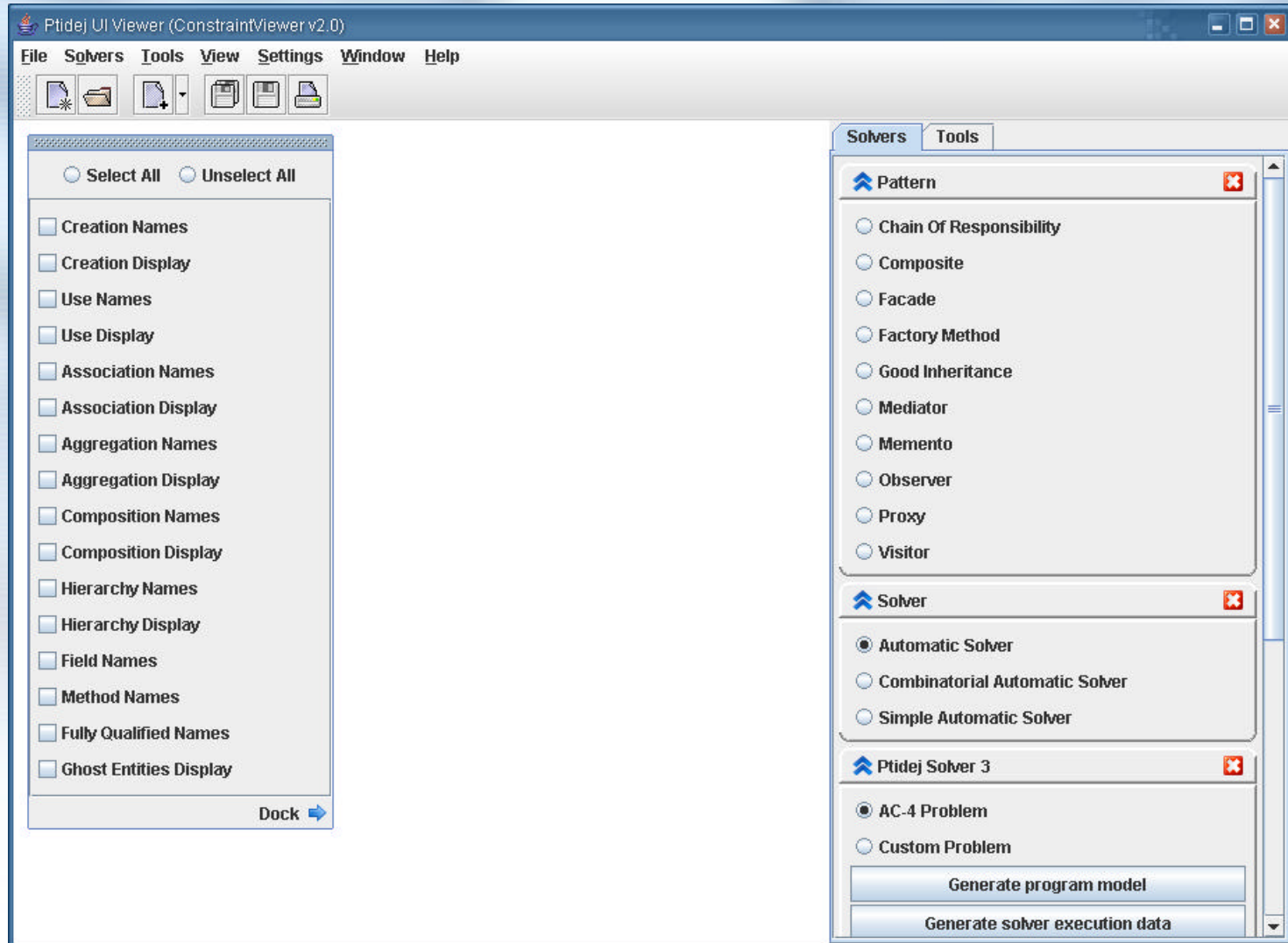
!

Ergonomique

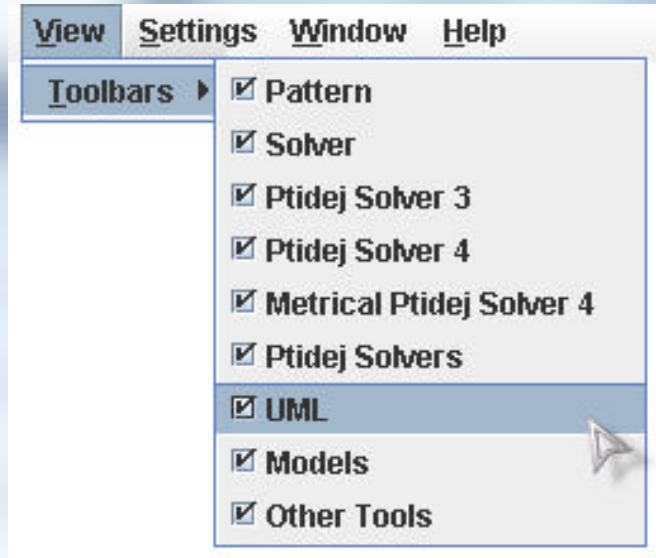
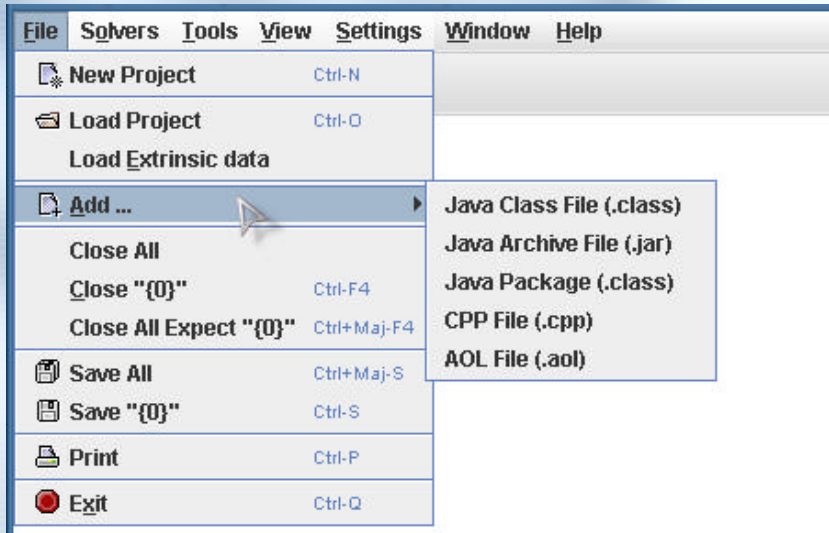
?



Comparaison - APRÈS (v2.0)



Menus (1)



Accès par groupe

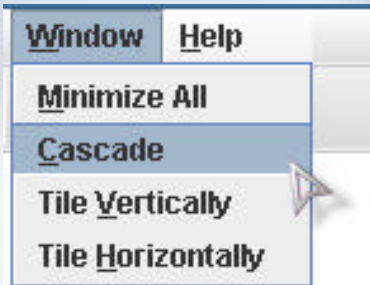
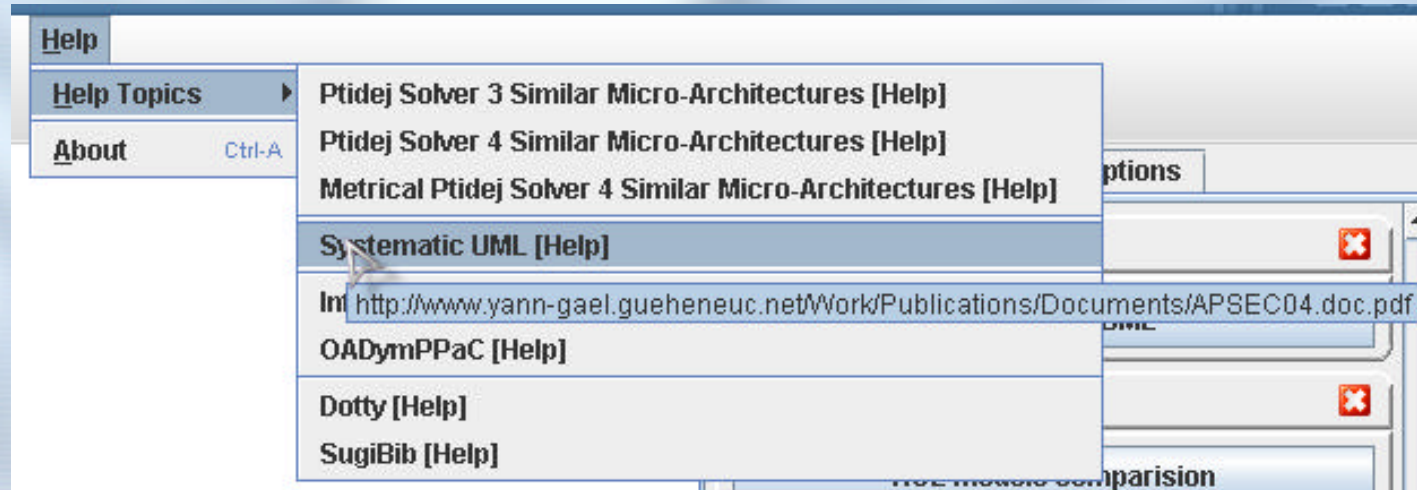
Possibilité de
personnalisation

(Ajout + Suppression
de barres d'outils)

Menus (2)

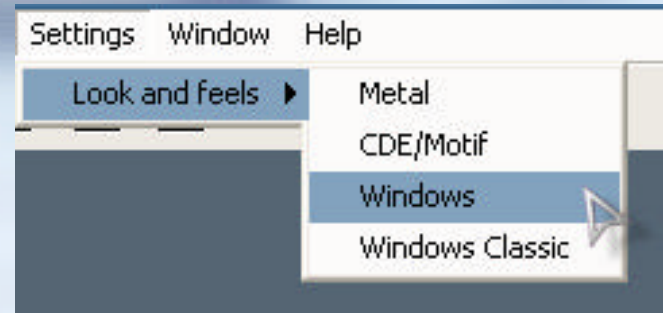
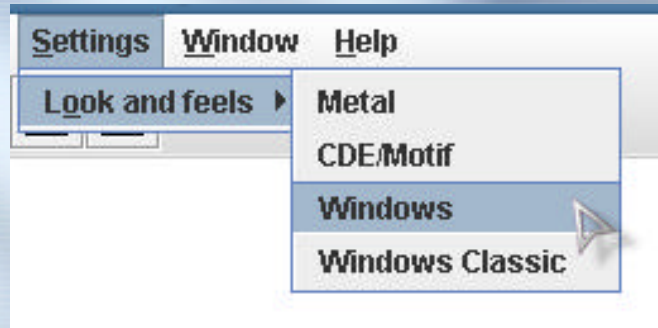


Aide rapide
pour les
outils
disponibles



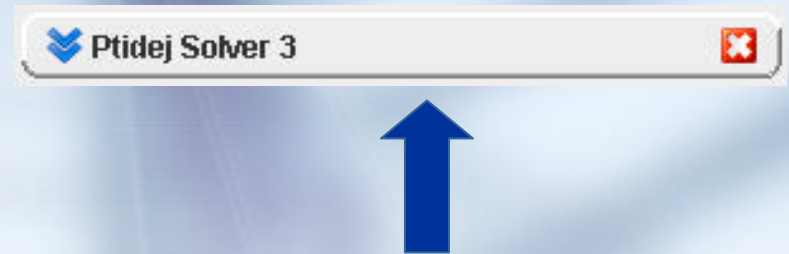
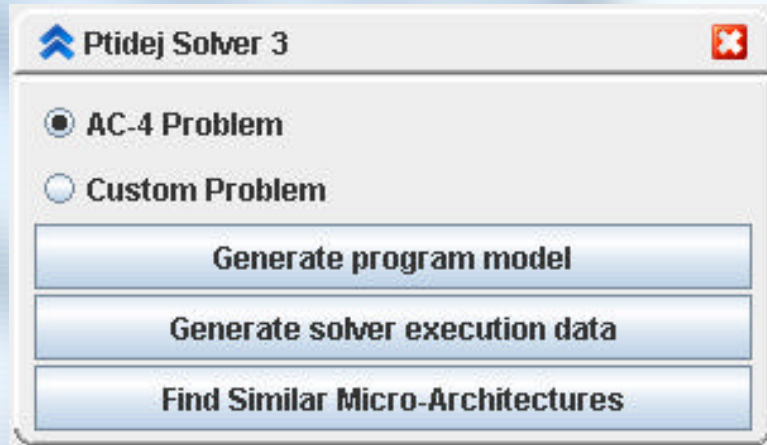
Facilités d'affichage

Multi Thèmes



Plusieurs Thèmes visuels
disponibles selon les goûts et le
confort de l'utilisateur

Personnalisation (1)

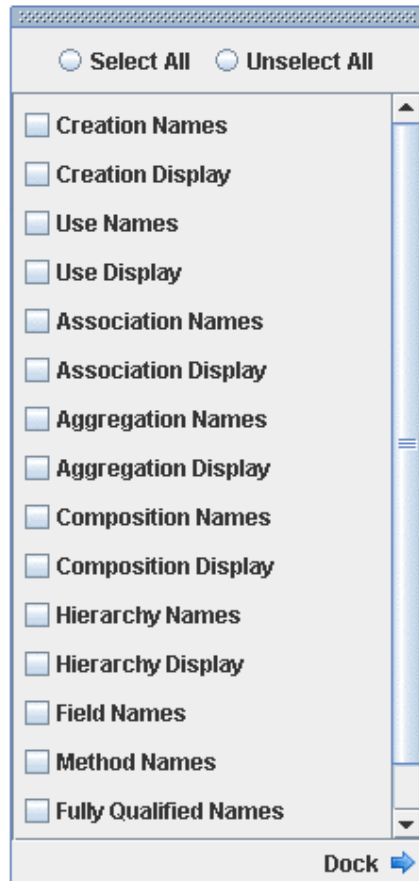


Possibilités de pouvoir dégager

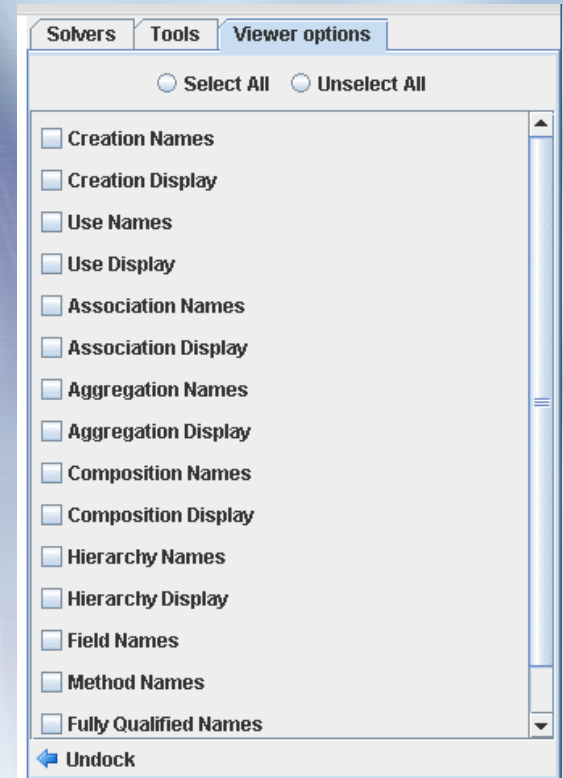
Des vues de travail non requises à l'instant

- Pourquoi confondre l'utilisateur avec des fonctions qui n'utilisera pas souvent ?
- Mais pourquoi, les mettre difficiles d'accès, alors qu'il en aura besoin un jour ?

Personnalisation (2)



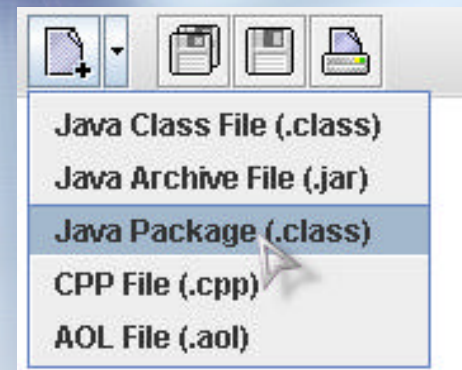
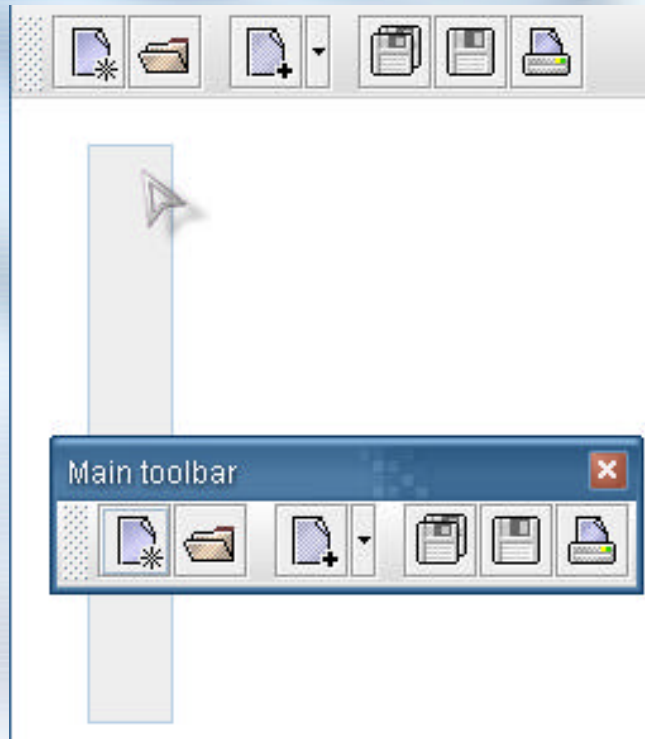
Pourquoi ne pas
avoir accès aux
commandes de
contrôle pendant
que l'utilisateur
change l'affichage
de son modèle ?



Transformation d'outil d'affichage

En palette en libèrent les panneaux de contrôle

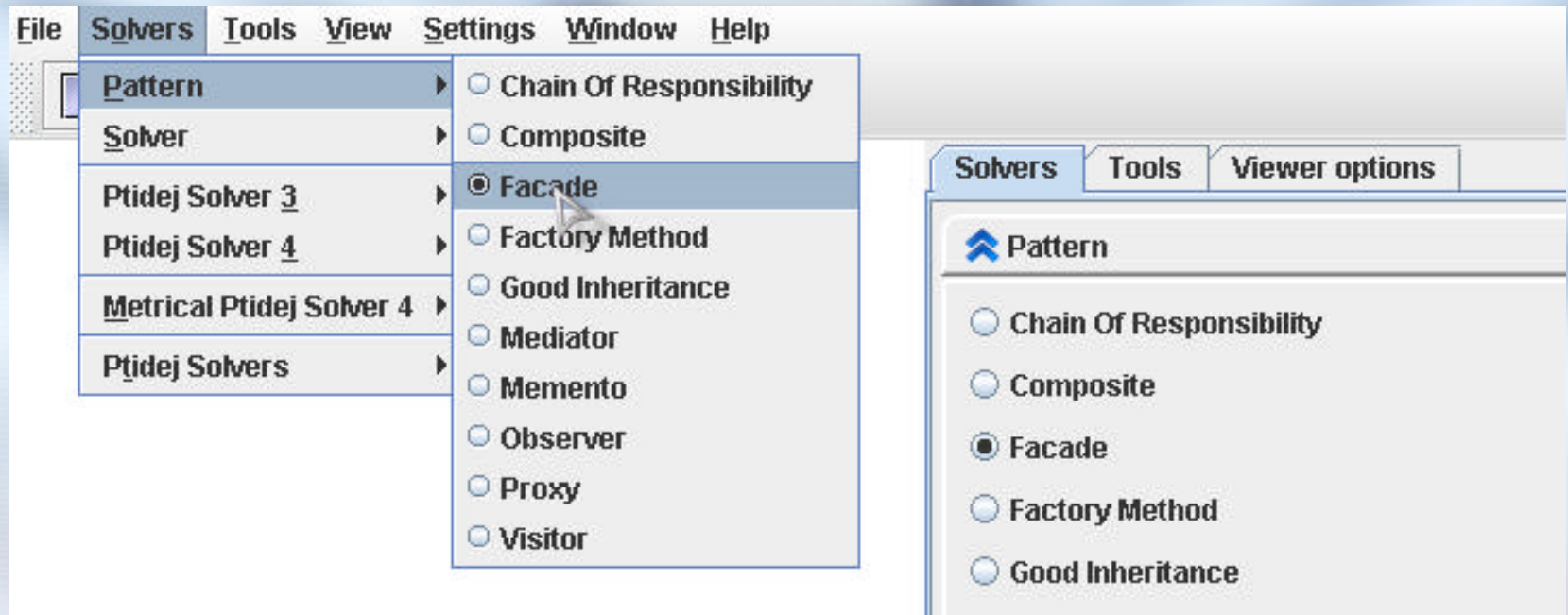
Personnalisation (3)



Un "File Menu" doit toujours être facile d'accès ! (Primordial)

Barre d'outils facile d'accès pour les fonctionnalités primaires

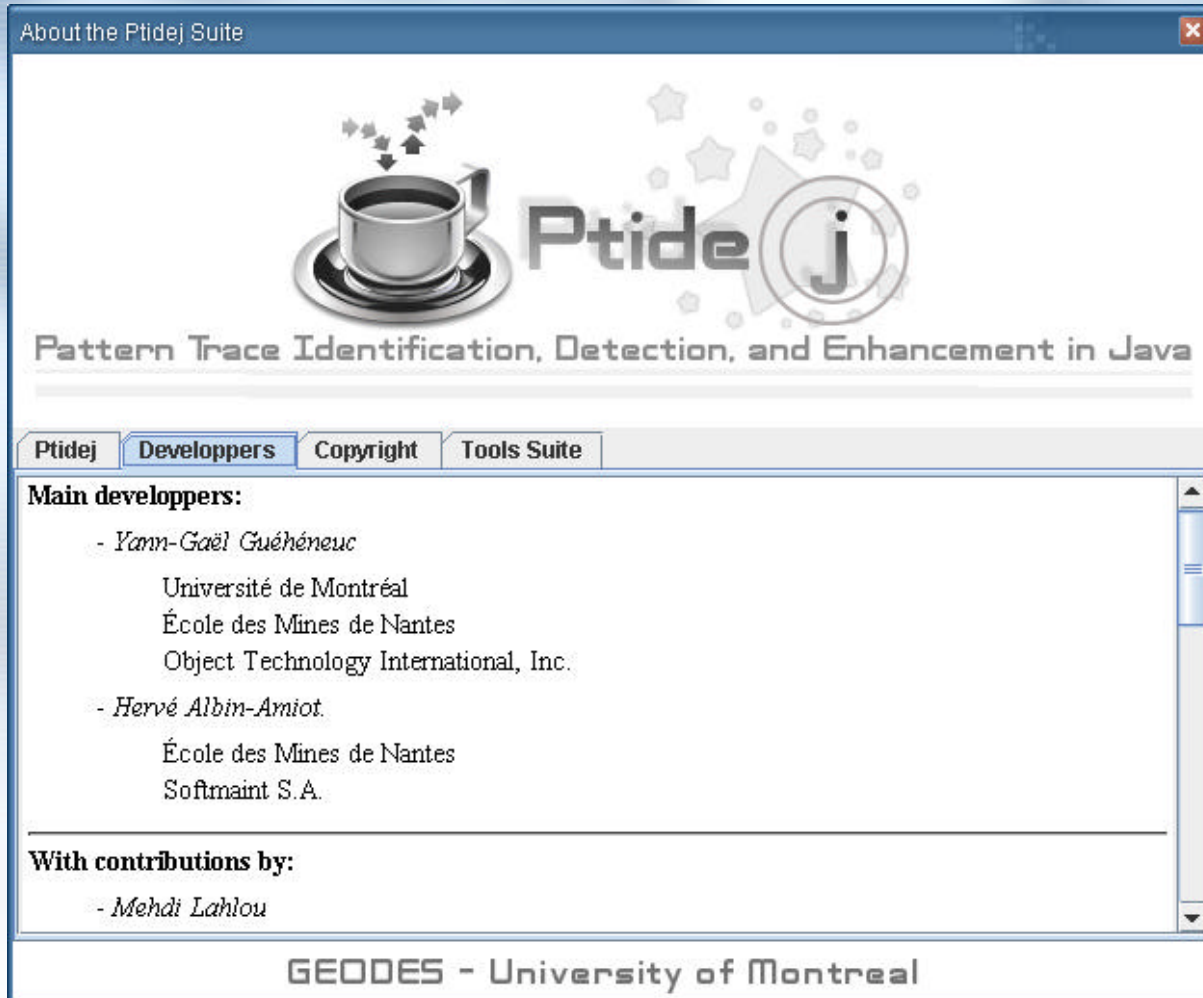
Personnalisation (4)



Synchronisation des différentes sources d'actions

Groupées pareillement

À Propos



Présenter clairement
les auteurs, la
licence, ainsi que les
version des
différents outils
intégrés

- Implantation d'une architecture pour supporter le multilingue.
 - Pour ajouter une langue, il suffit de créer ainsi un fichier contenant les ressources dans la langue désirée et le tour est joué.
 - Si le fichier de langue correspondant aux propriétés de l'OS est disponible, l'application démarre en utilisant ce dernier.
- Le cas échéant la langue se rapprochant le plus qui soit disponible est utilisée
(fr_CA vs fr_FR, en_CA)



- Architecture pour la gestion du multilingue fortement découplée du reste de l'interface graphique.

? Maintenance plus aisée.

Aujourd'hui, la maintenabilité est un facteur très important autant pour le noyau de l'application que pour l'interface graphique.

J'ai donc mis l'accent sur l'architecture de l'interface pour qu'elle soit maintenable facilement.
(MVC . Patrons de conceptions ...)



Création de composantes à partir d'une CLÉ

Ex: `addButton("Key");`

les différentes propriétés ainsi que
l'action désirée seront mis en place
automatiquement.

Mise en place d'un mécanisme automatisant
la multitudes de sources d'événements
(Toutes sources d'un même événement
connaissent une action d'événement
unique)

Modification pour Internationalisation



- La suite d'outils Ptidej compte au dessus de 1700 classes. En premier lieu, il fallait passer ces 1700 classe au peigne fin pour extraire tout ce qui était internationalisable, est le remplacer par une clé, qui sera identifiée dans le "resourcebundle" .

Tâche qui a consommée un temps non négligeable.

Ce fût un projet intéressant, mais reste du travail à faire (Amélioration, ajout de fonctionnalités pour extraire les préférences de l'utilisateur, Gestion multi-moniteurs ...).

Les utilisateurs sont de plus en plus exigeants sur l'usabilité, l'ergonomie et le design des interfaces graphique.

Remerciements

- En premier lieu, je tiens à remercier Mr Yann-Gaël Guéhéneuc pour son support et sa disponibilité.
- Mme Sung-Hui Park et, Mr Stefan Monnier
Pour leurs supervision.