

Trolltech DevDays 2006 : Qt Model-View-Controller Classes

par Matthieu Brucher (<http://matthieu-brucher.developpez.com/>) (Blog)

Date de publication : 01/11/2006

Dernière mise à jour :


L'utilisation des classes xModel et xView, résumé de la conférence de Bo Thorsen lors des Trolltech DevDays 2006.

- I - Qui est Bo Thorsen ?
- II - Résumé de la présentation
- III - Conclusion

I - Qui est Bo Thorsen ?

Bo Thorsen est consultant chez Thorsen Consulting, une entreprise spécialisée dans le consulting de Qt.

II - Résumé de la présentation

Le  **pattern MVC** est un motif qui permet de découpler les données d'un affichage. Qt4 propose plusieurs classes permettant d'afficher des widgets dans des vues, comme `QTreeView`, ...

L'avantage d'utiliser directement ce pattern et les classes de Qt qui l'implémentent est le gain de temps à long terme. Il est vrai qu'à priori, l'utilisation de ces classes est complexe, et il est plus simple de partir sur des `QxWidgets` que sur des `QxView`, mais à terme, cela porte ses fruits. Pour Bo Thorsen, ces classes sont les meilleures de la bibliothèque !

Le programmeur doit créer son propre modèle, la vue est gérée par Qt, et on verra ce qu'il en est du contrôleur. Dans le modèle, chaque donnée possède un index que la vue utilisera. Ces index comportent 3 niveaux d'information : un ligne, une colonne et un parent. Ce dernier est utilisé pour une vue hiérarchique, mais pas pour une vue dans une table, par exemple. Pour créer un modèle spécial, il suffit de surcharger 5 fonctions - au moins, on peut en surcharger plus si le besoin s'en fait sentir - qui sont **columnCount**, **rowCount**, **index**, **parent** et **data**.

A partir de ces modèles, on peut les envoyer dans une vue. Mais on veut parfois aussi filtrer et trier ces données avant la vue, et c'est le rôle des contrôleurs. Pour Qt, il s'agit simplement de proxys sur des modèles précédents. Le proxy classique est `QSortFilterProxyModel`. C'est donc un modèle qui sert de proxy, qui peut trier - en surchargeant **lessThan** -, mais aussi filtrer grâce à **setFilterX**, **filterAcceptRow** ou **FilterAcceptColumn**.

Maintenant, il est aussi possible de faire d'autres choses avec ce pattern. Par exemple, on peut utiliser le drag&drop rapidement. Un élément de données - donc toute la ligne - sera alors encodée en Mime - grâce à la fonction **mimeData** -, il suffira après de décoder et d'insérer les nouvelles données dans le modèle lors de l'appel à **dropMimeData**.

III - Conclusion

D'autres particularités de cette partie de Qt ont été discutée dans une autre présentation à laquelle nous n'avons malheureusement pas participé.

Qt4 propose désormais une manière de consulter des données rapidement. Pour développer un système à base de Model-View, le temps initial est long, mais le bénéfice intervient rapidement. Pour plus de renseignement, **une page d'aide** sur le site de Trolltech est proposé.

