



Une nouvelle plateforme ouverte pour l'édition et l'analyse de modèles UCM

Jason Kealey, Etienne Tremblay, Jean-Philippe Daigle,
Jordan McManus, Olivier Clift-Noël

{jkealey, etremblay, jpdaigle, jmcmanus, oclift}@SoftwareEngineering.ca

et

Daniel Amyot

damyot@site.uottawa.ca

Université d'Ottawa,
École d'ingénierie et de technologie de l'information (EITI)
Ottawa, Ontario, Canada

<http://www.site.uottawa.ca>

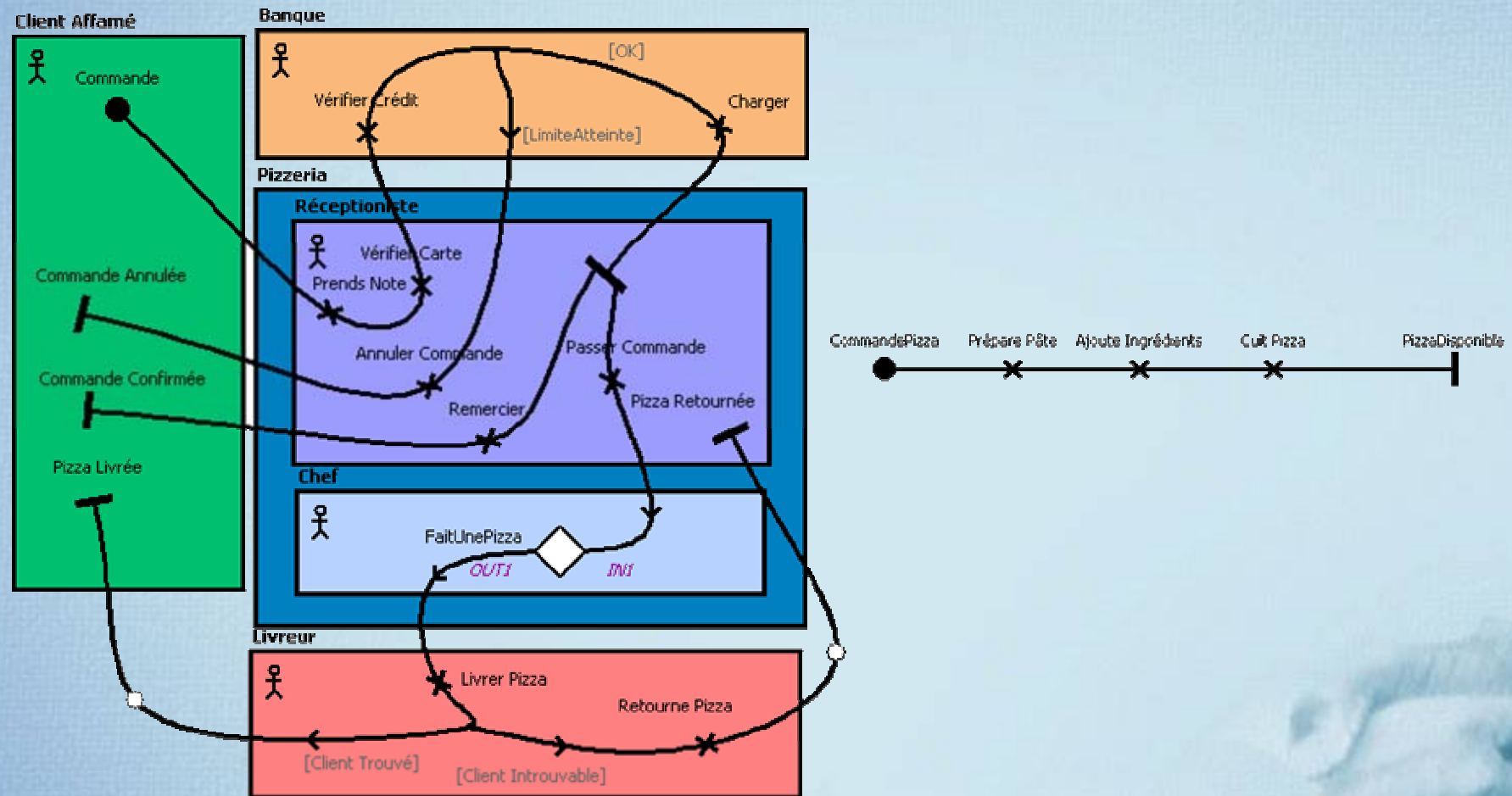


La notation Use Case Maps (UCM)

- **Scénarios graphiques**
 - Expriment l'architecture et le comportement du système
- Composante de la **User Requirements Notation (URN)**
- Entre autres, cette **notation versatile** a été utilisée pour:
 - La **modélisation et la validation d'exigences** et de processus d'affaires
 - La **génération de modèles de performance**
 - La **génération de scénarios** pour la conception détaillée
 - La **génération de tests**
 - La **visualisation de programmes existants** dans un contexte de rétro-ingénierie



Exemple de Use Case Map





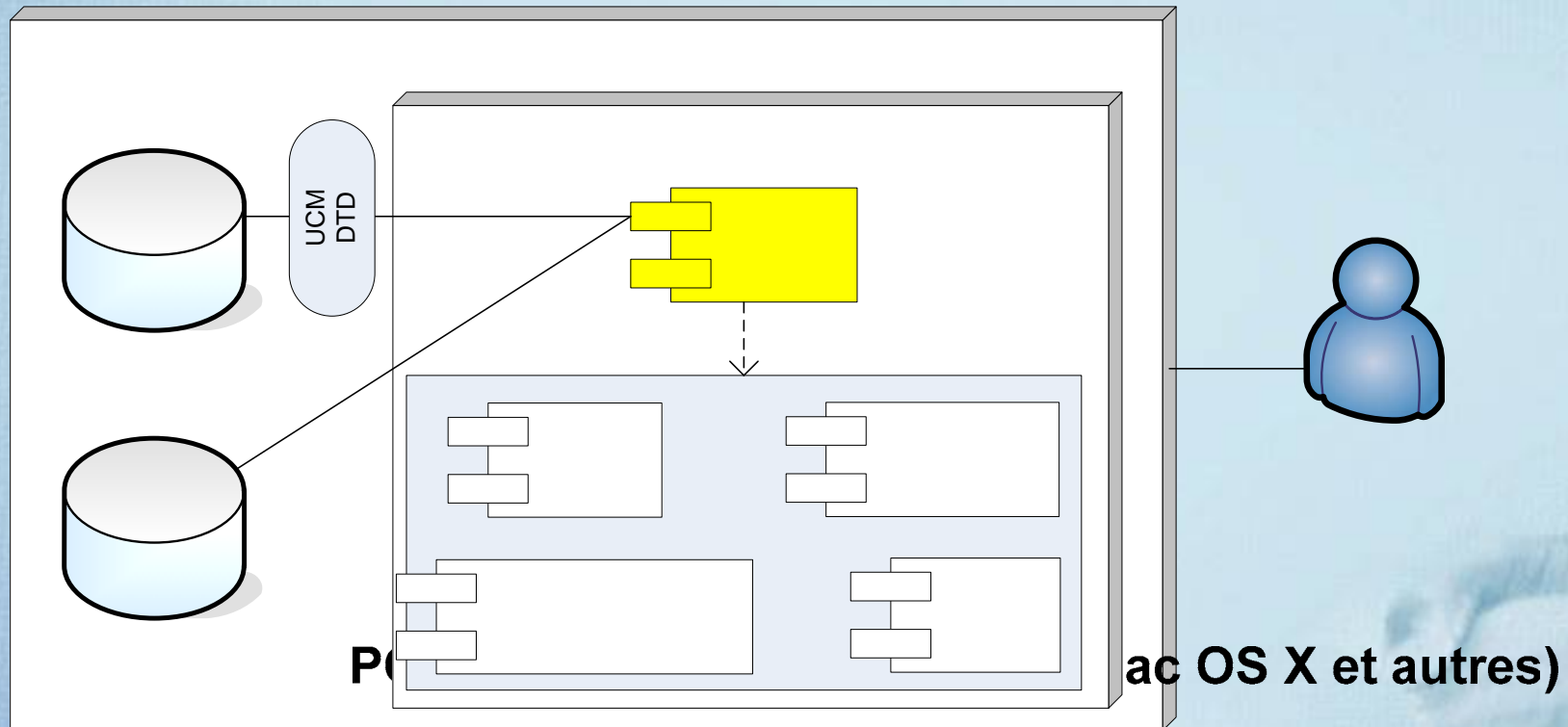
UCM Navigator: Problèmes

- **UCMNav**: 70 000 lignes de code C++
- Problèmes majeurs
 - Loin d'être **convivial**
 - **Déploiement** difficile
 - **Maintenance**, robustesse, extensibilité
 - 8 années de modifications par des étudiants ont effacé l'architecture
 - **Documentation et tests** presque absents



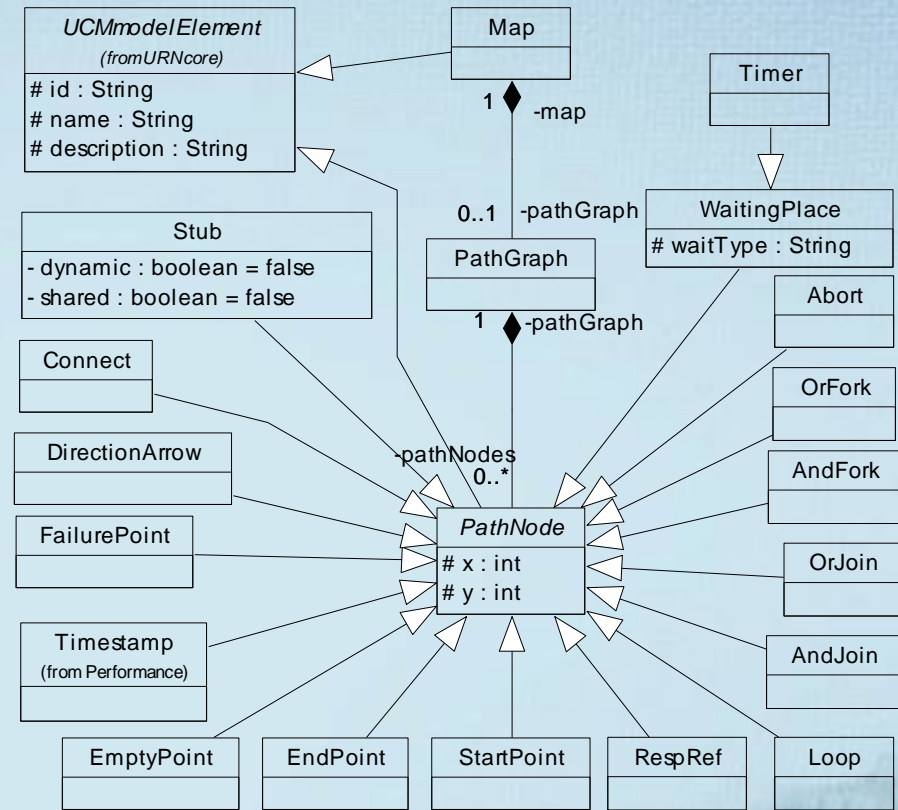
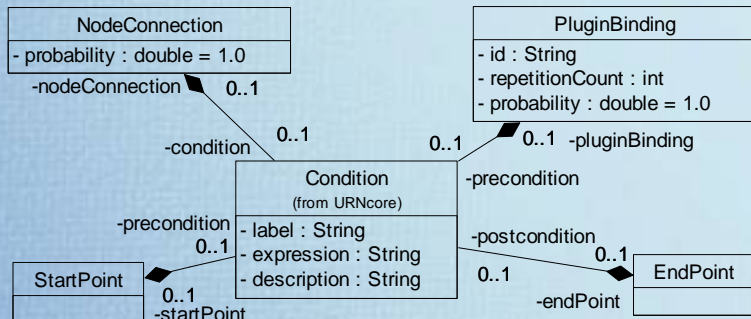
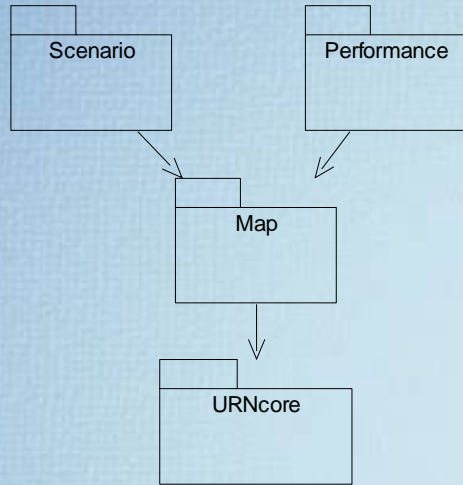
Architecture de jUCMNav

- jUCMNav:
 - **Plugin Eclipse** pour remplacer UCMNav

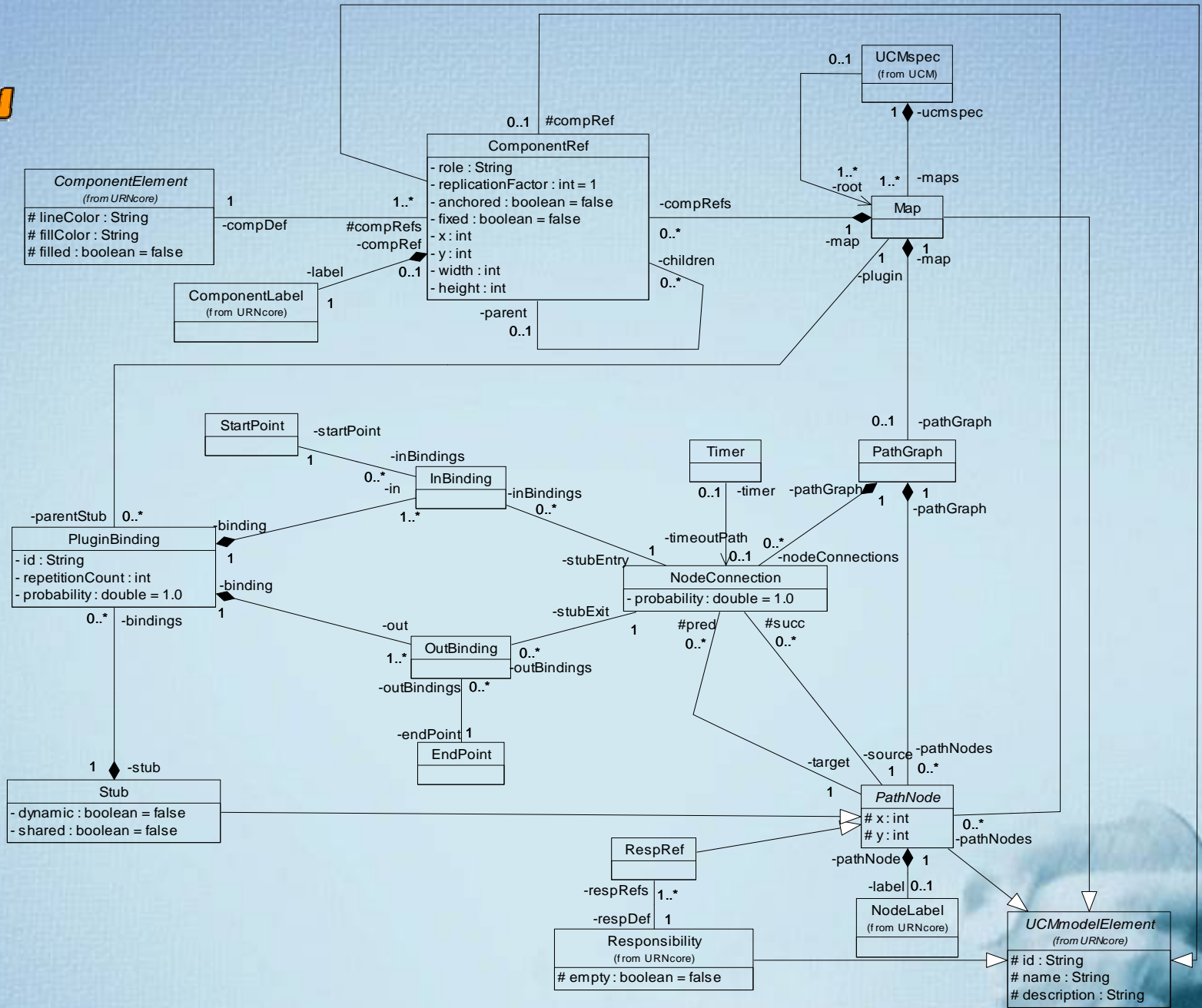




Méta-modèle Use Case Map



J. Kealey, E. Tremblay, J.-P. Daigle, J. McManus, O. Clift-Noël et D. Amyot
 Université d'Ottawa, Ontario, Canada



J. Kealey, E. Tremblay, J.-P. Daigle, J. McManus, O. Clift-Noël et D. Amyot
 Université d'Ottawa, Ontario, Canada



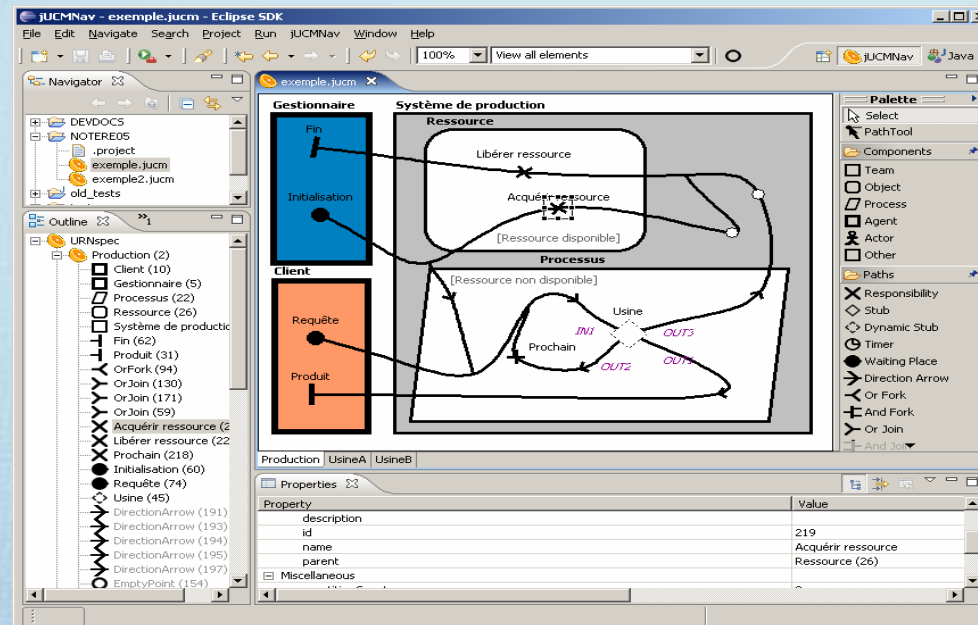
Eclipse Modeling Framework (EMF)

- **Génération automatique de code**
 - De modèles UML
 - 50 000 lignes de code EMF pour jUCMNav
- **Interface EObject et Notifier**
 - **Grande utilité pour réflexion et patron MVC**
- **Références bidirectionnelles**
 - L'implémentation EMF réduit la charge sur le développeur



Graphical Editing Framework (GEF)

- Facilite la création d'éditeurs graphiques
- Utilise le patron de conception MVC
 - Nous utilisons EMF pour la partie modèle





GEF/EMF : Exemple avec propriétés

- **L'utilisateur sélectionne un *EditPart***
 - *(partie contrôleur dans MVC)*
- GEF demande les **propriétés** du *EditPart*
- Les propriétés **sont construites en utilisant la réflexion** fournie par EMF
- Peu de nouveau code est nécessaire
 - Grâce à Eclipse, GEF et EMF



Fonctionnalités transférées

- **Support des éléments de base** et associations (*Bindings*)
- **Prévention de diagrammes syntaxiquement incorrects**
- Courbes (*B-splines*) reliant les nœuds
- Édition par menus contextuels pop-up
- Associations entre stubs et sous-diagrammes UCM
- Déplacement et effacement de groupe d'éléments



Fonctionnalités transférées (suite)

- Agrandissements (*zoom*), barres de déroulement
- Sauvegarde/chargement de modèles UCM
- Exportation des images (.bmp et .jpg au lieu de .eps)
- **Disponibilité multiplateforme (via Eclipse)
testée sur GNU/Linux, MS WindowsTM et Mac
OS XTM**
- Aide en ligne



Nouvelles Fonctionnalités

- **Annuler/rétablir** (*undo/redo*) à niveaux multiples
- **Placement automatique des éléments** (*auto-layout*) basé sur *GraphViz*
- Conception de diagrammes par **glisser-déposer**
- **Vue d'appoint hiérarchique** (*Outline*) éditable
- **Rétroaction lors de manipulations du diagramme**



Nouvelles Fonctionnalités (suite)

- **Étiquettes** (noms) **déplaçables et éditables** directement sur le diagramme
- Interface utilisateur suivant les normes Eclipse et utilisant les services de l'environnement de fenêtrage du système d'exploitation de l'utilisateur
- **Suite de tests fournie avec le code**, accompagnée de l'architecture permettant d'exécuter ces tests et ainsi d'assurer une certaine robustesse dans l'application
- Plus d'une trentaine de **documents architecturaux et d'implémentation** permettant d'adoucir la courbe d'apprentissage



Démonstration

- <http://www.SoftwareEngineering.ca/jUCMNav>
- Prérequis:
 - Eclipse 3.1 (<http://www.eclipse.org>)
 - GEF 3.1 (<http://www.eclipse.org/gef>)
 - EMF 2.1 (<http://www.eclipse.org/emf>)
- Optionnellement:
 - Graphviz dot (<http://www.graphviz.org>)



Difficultés rencontrées

- **Courbe d'apprentissage Eclipse/GEF**
 - Difficile mais valorisante
- **Sémantique de la notation UCM**
 - *Difficultés reliées à l'effacement et aux boucles*
- **Tests automatisés**
 - Habituellement simple, mais intégration continue avec CruiseControl non triviale.



Surprises plaisantes

- **Génération de code EMF**
- Généralisations
- **Passage de 3.0.1 à 3.1**
- Performance
- **Bonne architecture**
- Outils libres
- **Développement itératif**



Travaux futurs

- Support pour **responsabilités dynamiques**
- Distinction entre classes et instances de composantes
- Lecture/écriture du **format de fichier UCMNav**
- Lecture/écriture du format de fichier URN (XMI)
- **Recherche** dans un modèle
- Support des **définitions de scénarios**, traversée du modèle UCM et génération de scénarios individuels (avec sérialisation XML)



Travaux futurs (suite)

- **Intégration avec Telelogic DOORS** via des scripts et une librairie DXL
- Génération de rapports (en HTML et/ou PDF)
- Support des **annotations de performance** et génération de modèles CSM
- Support de la **notation GRL** (autre partie de URN) et de liens de traçabilité entre GRL et UCM
- Création de **points d'extensions Eclipse** afin que d'autres plug-ins puissent étendre jUCMNav



Conclusion

- Nous vous avons présenté l'outil jUCMNav, **une nouvelle plateforme ouverte pour l'édition et l'analyse de modèles UCM**
- L'outil peut être téléchargé ici:
 - <http://www.SoftwareEngineering.ca/jUCMNav>
- Contact:
 - jUCMNav@SoftwareEngineering.ca