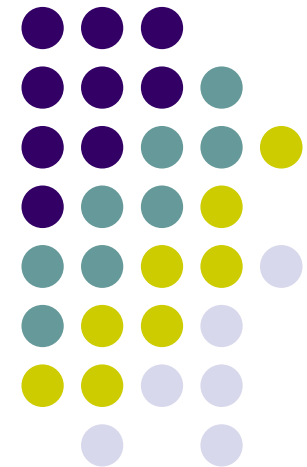


# GWT UML

## Un modeleur UML en GWT

---

Présentation de « mi-parcours »  
du PFE de Florian Mounier  
chez Objet Direct





# Rappel du sujet du PFE

- Conception et développement d'un modeleur de diagramme UML en JAVA
- En utilisant le Framework GWT
- En environnement open source
- A partir d'un prototype réalisé au préalable



# Contexte du projet

- Projet utilisant des technologies récentes
- Open Source
- Projet qui sera interfacé avec un autre projet réalisé par Henri Darmet
- S'appuie sur un prototype réalisé par celui ci
- Repose sur un autre projet d'Objet Direct : Tatami



# Technologies utilisés

- Développement sous Eclipse Ganymède
- Technologie : JAVA - GWT
- Gestionnaire de version : Subversion (SVN)
- Hébergement des sources et de l'application: Google Code
- Gestion de projet : GanttProject



# Google Web Toolkit

- Framework de développement initié par Google en 2006
- Permet de développer en JAVA
- Et d'obtenir une application Web AJAX utilisant le JavaScript, le HTML, et le DHTML.
- Permet de palier aux faiblesses du JavaScript lors du développement et du maintien du code de grosses applications Web
- Permet aussi le débogage dans l'éditeur JAVA



# Tatami

- Librairie open source permettant d'utiliser les fonctionnalités de DOJO à travers GWT
- DOJO est l'une des bibliothèques JavaScript les plus aboutie

# GWT UML Prototype



- Début du projet à partir de ce [prototype](#)



# Travail réalisé 1/3

- Mise en place de la plateforme de développement
- Mise à jour des librairies utilisées
- Création du site sur Google code
- Mise en place de la possibilité de pouvoir accéder à toutes les versions de l'application à travers Google code
- Immersion dans le code existant et réorganisation de celui-ci
- Abstraction de la couche graphique en utilisant le design pattern : bridge
- Intégration d'une autre librairie graphique : GWT Incubator Canvas





## Travail réalisé 2/3

- Ajout d'une IHM pour l'édition dynamique du diagramme
- Avec ajout d'un menu contextuel
- Ajout des animations de chargement
- Nettoyage du code, remise en forme
- Réécriture de plusieurs parties de code en vue de la réutilisation (pour les autres type de diagrammes notamment)
- Correction de nombreux bugs



## Travail réalisé 3/3

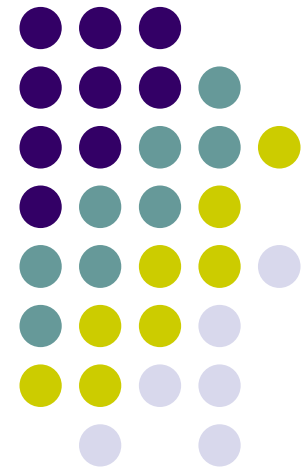
- Ajout des fonctionnalités manquantes :
  - Edition de tous les objets (quasiment)
  - Support de l'historique
  - Meilleure intégration dans Google Chrome
  - Ajout des raccourcis clavier
  - Ajout de la suppression des objets
  - ...

# Etat actuel

---

[Lien vers la dernière version](#)

[Lien vers version plus stable](#)





# Travail à réaliser

- Finir la réécriture tout en mettant en place les dernières fonctionnalités de base
- Terminer l'IHM en faisant en sorte que celle-ci soit la plus intuitive possible
- Réalisation de la JavaDoc
- Réalisation d'une couche d'abstraction des entrées souris / clavier afin d'offrir la possibilité aux personnes s'interfaçant à cette application de la tester.
- Implémentation du diagramme d'objet pour l'interfaçage avec la future application d'Henri Darnet

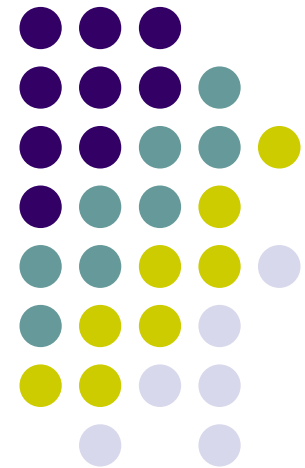


# Idées personnelles

- Si la durée du stage le permet voici quelques idées à implémenter :
  - Ajouter d'autres diagrammes UML (Séquence, Use Case...)
  - Export des diagrammes au format XML compatible avec d'autres modeleurs et export en format d'image
  - Sauvegarde des diagrammes à la manière des documents dans Google Docs (Mise en place d'une partie serveur)

# Planning

Diagramme de Gantt



# Questions ?

