



Département Informatique et Mathématiques Appliquées

Projet Long 2008

**ADMINISTRATION AUTONOME DE SERVEURS SUR
LA GRILLE AVEC UNE MACHINE VIRTUELLE**

Tutoriel d'utilisation de TUNe

Responsable : Daniel Hagimont - Professeur INPT/ENSEEIH - Daniel.Hagimont@enseeiht.fr

Co-encadrant : Laurent Broto – Etudiant en thèse à l'UPS - Laurent.Broto@irit.fr

Superviseur industriel : Emmanuel Murzeau - emmanuel.murzeau@airbus.com

Chef de projet : Ezequiel Geremia - ezequiel.geremia@etu.enseeiht.fr

Etudiants :

- Julien Louisy
- Julien Clariond
- Hery Randriamanamihaga
- Ezequiel Geremia
- Mathieu Giorgino

Sommaire

1 - Mise en place de l'environnement Système.....	3
1.1 - Environnement java.....	3
1.2 - Environnement Système pour un fonctionnement en local.....	3
1.3 - Environnement Système pour un fonctionnement en réparti.....	3
1.4 - Subversion.....	3
2 - Mise en place de l'environnement de Développement.....	4
3 - Mise en place de TUNe.....	5
3.1 - Récupération de l'application.....	5
3.2 - Execution du programme de démonstration.....	5

1 Mise en place de l'environnement Système

1.1 Environnement java

Pour fonctionner, TUNe a besoin de la version 6 de java ainsi que de la dernière version de Ant. Leur installation par apt-get se fait à l'aide de la ligne de commande suivante:

```
sudo apt-get install sun-java6-jdk ant
```

L'utilisation de la version 6 de java par le système se fait par la commande suivante :

```
sudo update-java-alternatives -s java-6-sun
```

1.2 Environnement Système pour un fonctionnement en local

TUNe nécessite le fonctionnement de ssh sans mot de passe sur le hôte local. Pour cela, il faut une installation de ssh fonctionnel ainsi que l'authentification sur le hôte local par clé RSA

La commande suivante permet d'installer un serveur ssh :

```
sudo apt-get install openssh-server
```

Vérifier ensuite que ssh fonctionne sans demander de mot de passe sur le hôte local en executant la commande suivante :

```
ssh localhost
```

Si un mot de passe est demandé, terminer la manipulation (avec Ctrl + C) puis mettre en place l'authentification ssh sur le hôte local par clé RSA.

- Générer une paire de clé RSA avec la commande suivante :

```
ssh-keygen -t rsa
```

- Ajouter la clé publique générée dans la liste des hôtes autorisés :

```
ssh-copy-id -i ~/.ssh/id_rsa.pub localhost
```

1.3 Environnement Système pour un fonctionnement en réparti

A Faire

1.4 Subversion

L'installation de TUNe nécessite également une installation fonctionnelle de Subversion. Vérifier que Subversion est installé sur la machine en tapant dans un terminal la commande suivante:

```
svn
```

Si la commande n'est pas reconnue, installer Subversion avec la commande suivante :

```
sudo apt-get install subversion
```

2 Mise en place de l'environnement de Développement

Le développement de composants TUNe nécessite l'environnement de développement intégré Topcased. Il est disponible en téléchargement à l'adresse suivante :

```
http://gforge.enseeiht.fr/frs/download.php/1047/org.topcased.rcp-  
linux-R-1.2.0-200712131010.zip
```

Une fois le téléchargement effectué, il reste à extraire le contenu de l'archive pour pouvoir lancer Topcased.

3 Mise en place de TUNe

3.1 Récupération de l'application

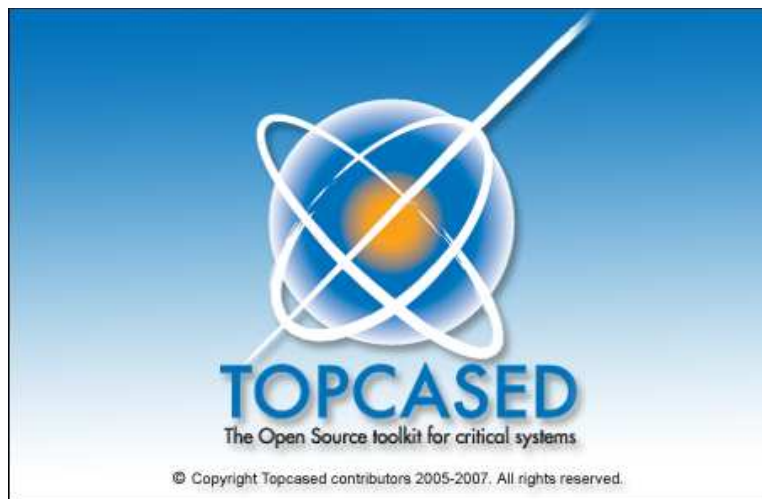
Récupérer TUNe depuis les dépôt Subversion avec la commande suivante :

```
svn co http://hagimont.svn.enseeiht.fr/TUNe/tags/stage-n7 --username=stagen7
```

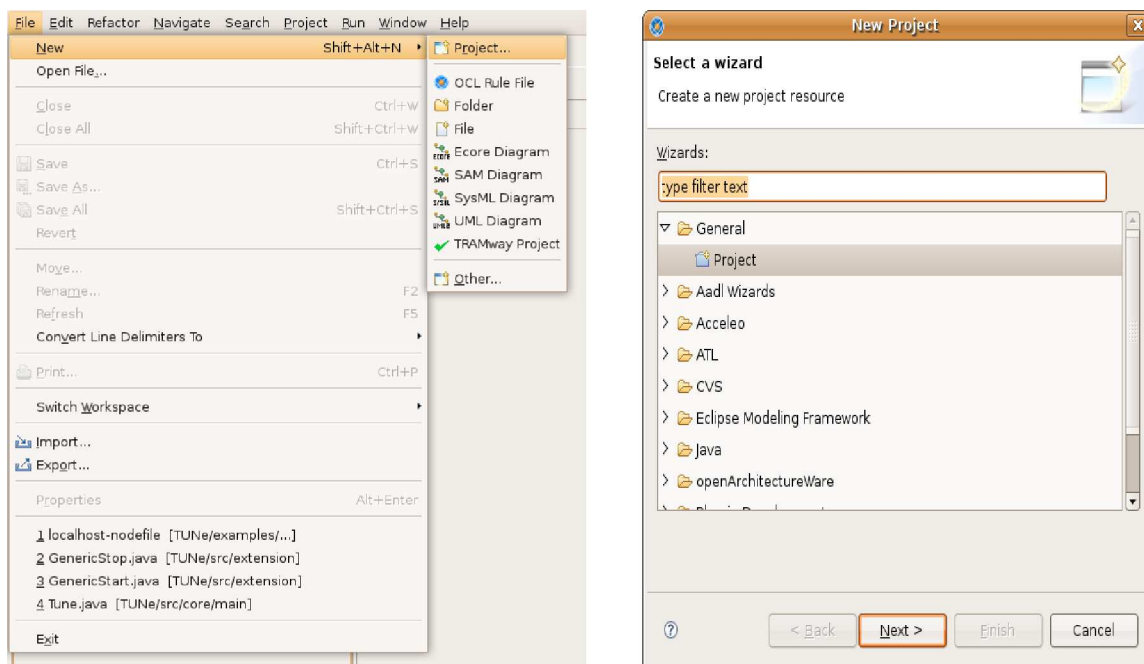
3.2 Execution du programme de démonstration en local

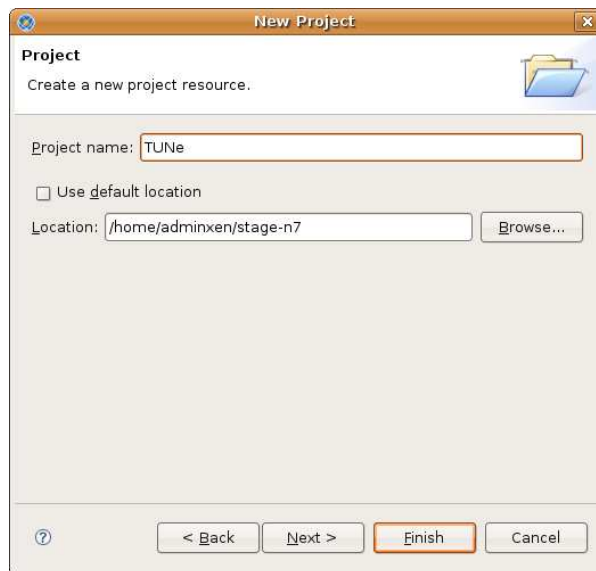
La configuration du démonstrateur AppliBete est nécessaire pour son execution. Cela se fait à l'aide de Topcased.

Pour commencer, lancer Topcased.



Créer un nouveau projet à partir des sources téléchargés depuis les dépôts:





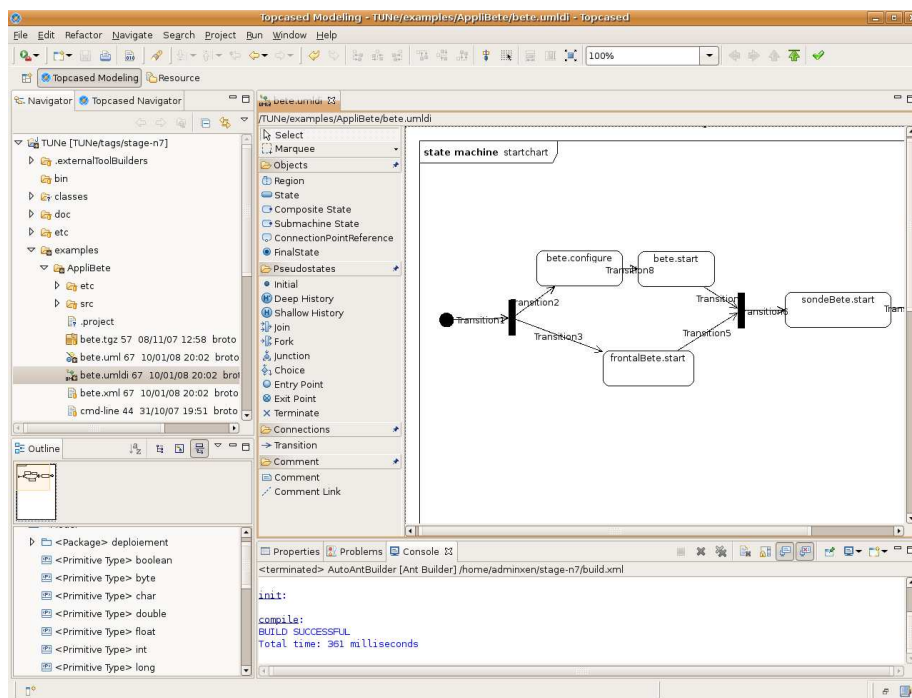
Ne pas utiliser l'emplacement du projet par défaut. <chemin de TUNe> correspond à l'emplacement où TUNe a été récupéré depuis les dépôts. Dans notre cas, le chemin correspond à

/home/adminxen/stage-n7

Terminer la création du projet (cliquer sur le bouton Finish)

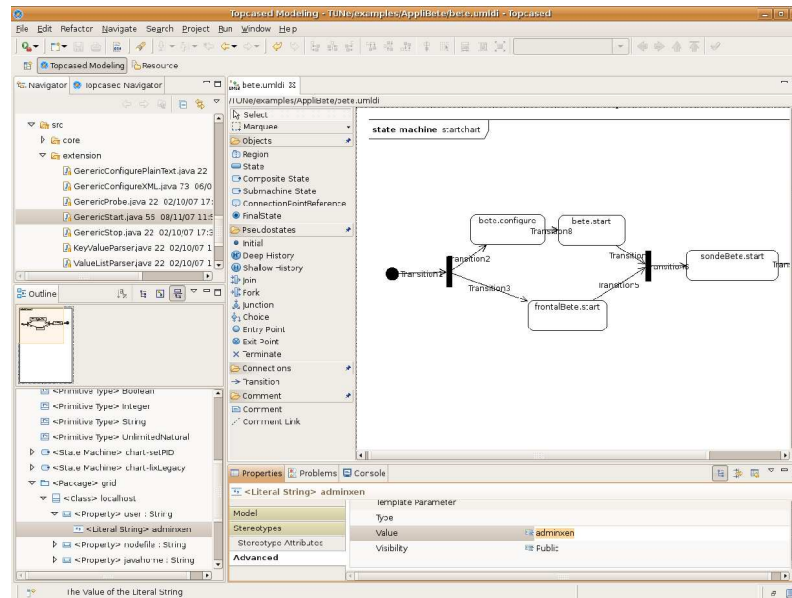
Ouvrir le diagramme de l'application de démonstration :

TUNe/examples/AppliBete/bete.uml



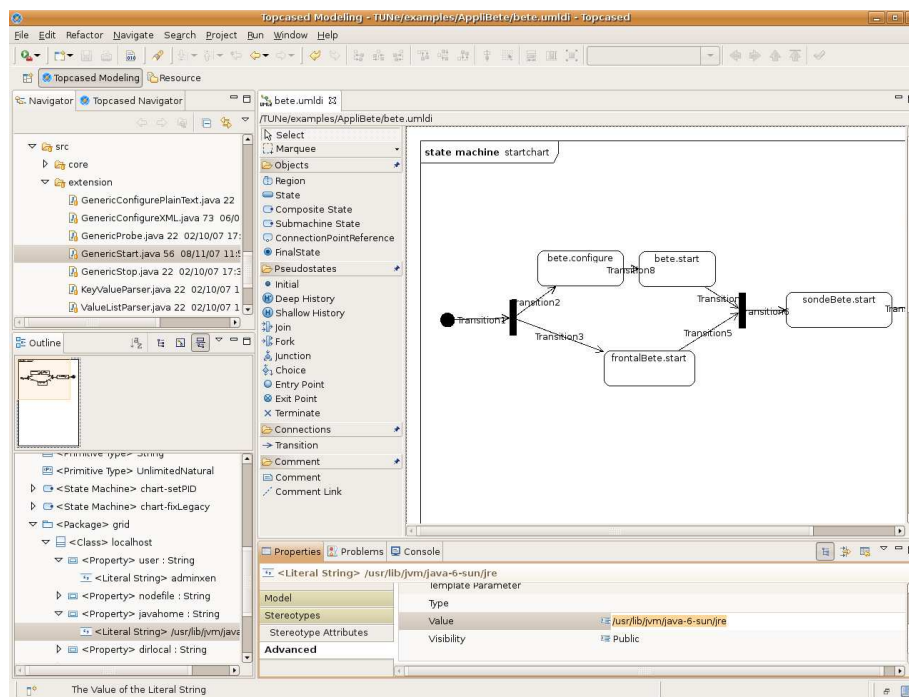
Modifier les propriétés de l'application afin de les rendre conforme à la configuration du système, à savoir, le nom d'utilisateur :

Dans le volet Outilne, sélectionner le <Literal String> du noeud <Package> Grid/<Class>localhost/<Property>user, puis dans le volet Properties, accéder aux options avancées pour éditer le champ value



Renseigner également l'emplacement de l'installation de java :

Dans le volet Outilne, sélectionner le <Literal String> du noeud <Package> Grid/<Class>localhost/<Property>javahome, puis dans le volet Properties, accéder aux options avancées pour éditer le champ value



Enregistrer avant de poursuivre.

Ouvrir le fichier `localhost-nodefile` du dossier `examples/AppliBete` pour y ajouter l'adresse du hôte local.

L'application peut maintenant être lancée : dans un terminal, se positionner dans la racine du projet TUNe

```
cd stage-n7
```

Compiler l'application :

```
ant clean compile dist
```

Lancer TUNe

```
ant tune
```

Pour déployer le démonstrateur, exécuter la commande suivante :

```
deploy examples/AppliBete bete.uml
```

Le déploiement assigne un numéro pour l'application.

Pour terminer la démonstration, exécuter la commande suivante :

```
undeploy <numero de l'application>
```

Pour quitter TUNe, saisir la commande suivante :

```
quit
```