

Présentation et initiation à l'Atelier B - Version 4

Lors de la première séance, vous allez découvrir la version 4 de l'Atelier B ; vous allez créer avec l'aide de l'enseignant, votre premier projet puis faire quelques manipulations des fonctionnalités de l'Atelier.

Atelier B (AB_V4) est une plateforme logicielle qui aide à la spécification et la construction de programmes en utilisant la méthode et le langage B.

La notion de **projet** structure toute activité avec l'atelier AB_V4 ; tout travail de spécification ou développement se fait donc dans le cadre d'un projet. En B, on distingue des projets de développement de logiciels (*Software development*), des projets de modélisation de systèmes (*System modelisation*) et de validation de données (*Data Validation*) .

☞ Créez sous Linux, votre **répertoire de travail pour les TD de B**, par exemple WS-TDB pour tous les TD¹. Attention, si les répertoires existants comportent des accents !

AtelierB n'aime pas les accents et autres caractères très spéciaux dans les noms des répertoires. Pour cela, en fonction de votre environnement de travail, créer un répertoire avec des noms sans accents (par exemple ln -s /home/moi/Réseaux/MonModule/WS-TDB /home/moi/WS-TDB).

Vous utiliserez le nom de répertoire sans accents dans les préférences de AtelierB, mais vos données seront bien sauvegardées ...

☛ Créez votre espace de travail (Workspace) ; le nommer et le lier au répertoire de travail (par exemple à WS-TDB) : AtelierB > New > Workspace Les projets seront créés dans cet espace. Attention à sélectionner ce workspace avant de créer vos projets (sinon ils sont perdus s'ils restent dans l'espace local par défaut).

☞ Vous devez **définir votre répertoire de travail** (par exemple MesProjetsB pour tous les projets B) en allant dans AtelierB > Preferences puis dans l'onglet Project. Par défaut, tous les projets que vous allez créer seront rangés dans ce répertoire de travail.

☛ Vous **devez personnaliser les entêtes** de projet (par exemple Prénom, Nom - IUT Info Nantes 2014 pour tous les projets B) en allant dans AtelierB > Preferences puis dans l'onglet New Components. Par défaut, tous les projets que vous allez créer auront le même format d'entête.

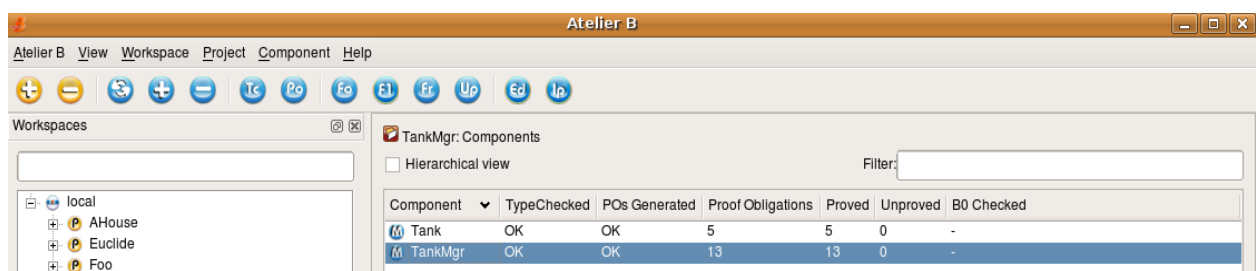


FIGURE 1 – Bandeau/menu principal de Atelier B - V4

Les projets (Project)

Nous démarrons par les projets de développement de programmes/logiciels (*Software development*). Tout projet doit être nommé ; un sous-répertoire de votre répertoire de travail MesProjetsB est associé à chaque projet. Les données de chaque projet sont rangées dans son propre répertoire. Ce répertoire va contenir

1. En cas de difficulté d'accès réseaux, cela peut être créé dans /tmp, mais attention à tout recopier avant de quitter !

les sources des spécifications B (machines `.mch`, raffinements `.ref`, implantations `.imp`) que vous écrirez dans le cadre de ce projet.

Dans le répertoire associé à chaque projet il y a deux sous-répertoires :

- `bdp` qui va contenir toutes les informations créées par l'atelier B pendant vos sessions,
- `lang` un répertoire qui va contenir les codes exécutables générés pour votre projet.

Remarque : par défaut tous ces (sous-)répertoires sont créés par l'atelier B.

Exemple : si vous créez et travaillez sur un projet nommé `TankMgr`, vous allez trouver un répertoire `TankMgr` à l'endroit indiqué comme répertoire de travail.

Lorsqu'on crée un nouveau projet, il est vide (de tout composant) ; on lui ajoute des composants (machines ou raffinements). Vous trouverez facilement ces fonctionnalités dans l'interface de l'atelier.

Dans l'atelier B, on peut ouvrir/fermer/archiver/supprimer un projet. Un projet peut en utiliser d'autres comme bibliothèques (voir les propriétés d'un projet).

Les machines et les raffinements (Component)

Un projet contient une ou plusieurs machines abstraites et un ou plusieurs raffinements. Toute machine abstraite est sauvegardée dans un fichier ayant le même nom que la machine (avec l'extension `.mch`) ; par exemple la machine `TankMgr` est sauvegardée dans le fichier nommé `TankMgr.mch`. Tout raffinement est sauvegardé dans un fichier ayant le même nom que le raffinement (avec l'extension `.ref`). Les machines peuvent être créées indépendamment puis ajoutées au projet sélectionné (avec `Components > Add`).

Les traitements/analyses élémentaires sur les machines

Après la saisie d'une machine abstraite dans un fichier, et après l'avoir ajouté à un projet, on peut lui appliquer des traitements : contrôle de type (fonction `Tc`) ; génération des obligations de preuve (fonction `Po`) ; preuve des obligations générées (fonctions `f0`, `f1`, et `lp` pour la preuve interactive).

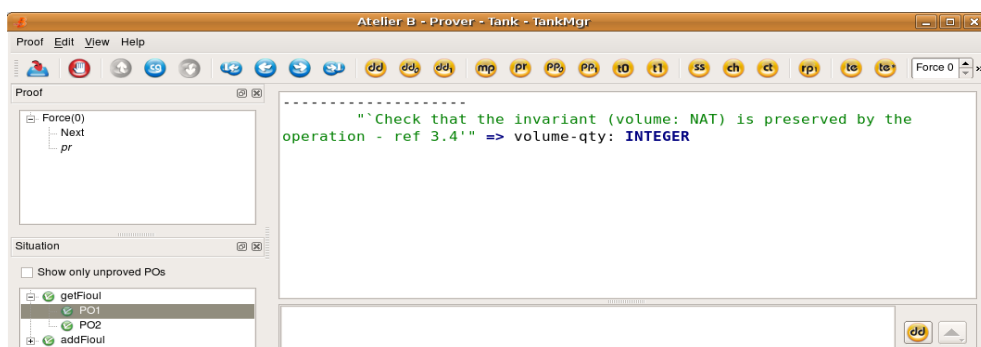


FIGURE 2 – Bandeau/menu principal de l'interface de preuve (de AtelierB-V4)

Pour la documentation des projets, on peut exporter un composant (machine/raffinement) vers les formats `.rtf`, `.pdf` ou `.tex` ; il faut utiliser pour cela la fonctionnalité `Component > Export as...`

Exercice : Exemple de projet

☛ Créez sous l'atelier B un projet nommé `Etudiant1` ; dans ce projet, créez une machine nommée `Etudiant`. éditer la machine `Etudiant`, compléter là et tester les fonctionnalités.

```
MACHINE
  Etudiant      /* le nom de la machine (ou modèle formel) */
SETS
  ETUDIANT      /* un ensemble abstrait */
; IDGROUPE = {g1, g4, g3, g2} /* un ens. enumere d'identifiants*/
VARIABLES      /* liste de toutes les variables utilisées */
  etudiants, idGroupes, groupeDe
INVARIANT
  etudiants    <: ETUDIANT /* un sous-ensemble */
& idGroupes    <: IDGROUPE /* un sous-ensemble d'identifiants de groupes*/
& groupeDe    : etudiants --> idGroupes /* une fonction */
INITIALISATION
  etudiants, idGroupes, groupeDe := {}, {}, {}
END
```