# Projet arbres binaires 1
# Etude formelle

licence informatique

20 mai 2004

## Table des matières


**1 Interface homme machine** **2**
- 1.1 Etude formelle . . . 2
- 1.2 Aperçu de l'interface graphique . . . 2

**2 Réalisation du projet** **2**
- 2.1 Difficulté et choix d'implémentation . . . 2
- 2.2 Les différents modules . . . 3
  - 2.2.1 L'onglet Création . . . 3
  - 2.2.2 L'onglet Exécution . . . 4
  - 2.2.3 L'onglet Paramètrage . . . 7

# 1 Interface homme machine

## 1.1 Etude formelle

L'IHM sera organisée avec des onglets contenant chacun les fonctions le concernant. On distingue :

- l'onglet création
- l'onglet exécution
- l'onglet paramétrage

Chacun de ces onglets sera développé conjointement par différent membres du groupe. C'est pourquoi, il faut que tous ces composants s'intègrent parfaitement à la fenêtre principale de l'application. Comme le langage utilisé est Java, il nous suffit d'utiliser les propriétés d'héritage ,de polymorphisme et d'encapsulation en général pour que les différentes entités soient totalement intègrées au projet.

Chaque onglet possèdera une barre d'outils dont le modèle sera défini dans un classe que chaque développeur pourra utiliser. Ainsi, bien que développé par différentes personnes, l'application sera homogène.

On distinguera plusieurs classes :

- une classe Fenetre qui se chargera d'organiser et de mettre en place tous les différents composants. Elle héritera de JFrame
- une classe ToolBar (barre d'outils) qui étendra JToolBar et qui spécifiera le modèle de la barre d'outils utilisée dans les différents onglets
- une classe Constantes abstraite ne contenant que des champs public final static dont tous les composants se serviront (cela contribuera à homogeneiser le projet)
- trois sous classes étendant toutes JPanel (création, exécution et paramètrage) et contenant les différents composants relatifs à leurs utilisations respectives.

Parallement sera développé un package permettant de créer facilement des arbres graphiquement. En effet, cela ne sera pas seulement utilisé dans l'onget creation mais aussi dans l'onglet exécution.

Voir diagramme de classes (Fig. 1 (page 3) ).

## 1.2 Aperçu de l'interface graphique

L'interface aura à peu près l'allure de la Fig. 2 (page 4) . On peut voir qu'elle est divisée en onglets et que chaque onglets est susceptible d'en contenir d'autres. On voit aussi que la barre d'outils se situera à gauche mais elle pourra être déplacé pour acqérir un placement non génant et pratique pour l'utilisation. Notons que les différents boutons présent dessus ne sont là qu'à titre d'exemple.

# 2 Réalisation du projet

## 2.1 Difficulté et choix d'implémentation

La principale difficulté réside dans le fait qu'il faut utiliser et intégrer à l'interface tous les algorithmes développés par le groupe de projet 2. C'est pourquoi le groupe 2 nous fournira une classe contenant tous les prototypes de leurs méthodes pour que nous puissions déjà prévoir leur utilisation future.

A ça s'ajoute une nouvelle difficulté qui est l'exécution pas à pas de certains algorithmes. On ne peut pas suivre le code compilé intruction par instruction (ce n'est pas un déboggeur dont il est question ici !). On a en fait besoin que le groupe de projet 2 implémente des méthodes retournant les résultats des algorithmes mais aussi une liste contenant toutes les étapes de cet algorithme. Ainsi nous seront en mesure d'afficher chaque étape en parcourant la liste (On pourrait aussi imaginer que le dernier élément de la liste est le résultat final de l'algorithme.

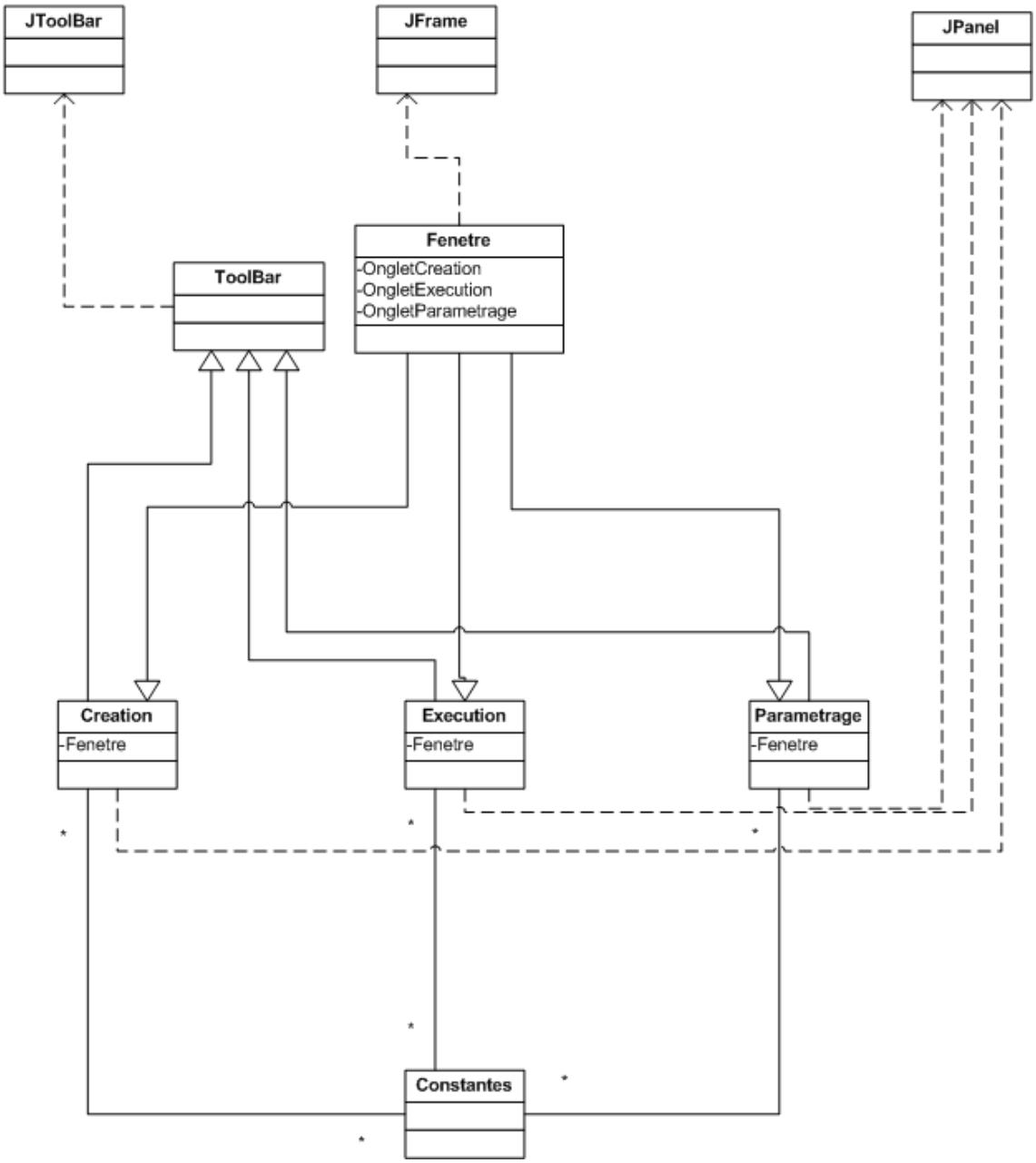


FIG. 1 – Diagramme de classes simplifié

La communication entre les deux groupes est donc primordiale pour que le projet aboutisse. Le schéma de la FIG. 3 (page 4) illustre cette communication.

## 2.2 Les différents modules

### 2.2.1 L'onglet Création

L'onglet création devra permettre de créer, d'importer ou de sauvegarder un arbre. On devra aussi pouvoir utiliser un arbre créé dans l'onglet création, dans l'onglet exécution.

La création se décompose comme suit :

- Création d'un sommet
- Création d'un fils gauche **et** d'un fils droit (les arbres sont parfaits (0 ou 2 fils)).
- Choix de labels (liste d'entiers et/ou de chaînes de caractères) pour les sommets et pour les arêtes.
- On doit pouvoir bouger les sommets d'un endroit à un autre avec les arêtes

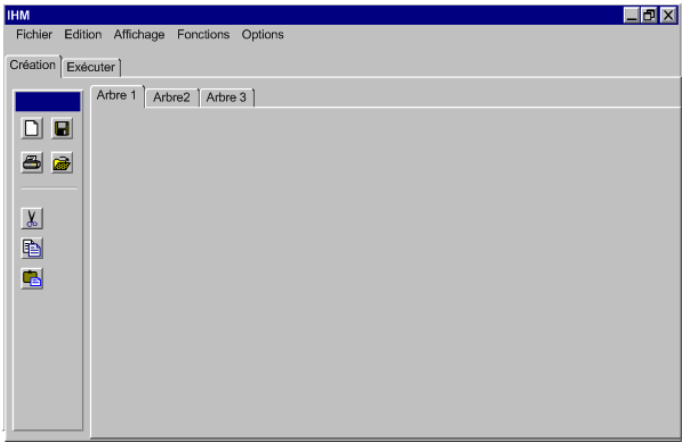


FIG. 2 – Aperçu de l'interface graphique

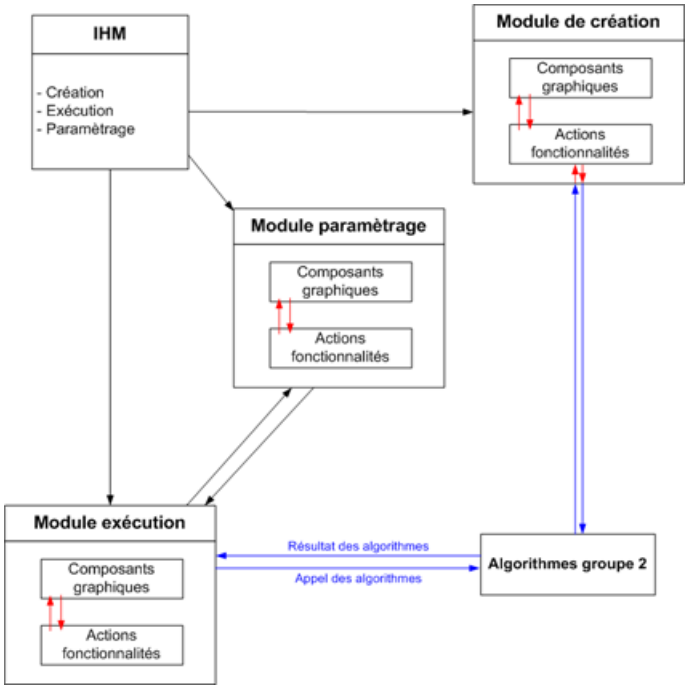


FIG. 3 – Intéractions entre les différentes entités du projet

– La fenêtre création doit contenir des ascenseurs permettant de faire défiler les zones dépassant la vue d'affichage (JViewPort en Java)

La FIG. 5 (page 5) et FIG. 4 (page 5) montrent ce que pourrait donner la création d'un arbre. On imagine également un menu popup qui permettrait de supprimer ou de modifier un noeud selon son désir comme à la FIG. 6 (page 5) .

### 2.2.2 L'onglet Exécution

L'onglet exécution doit pouvoir utilisé les arbres créés dans l'onglet création. Il doit aussi être capable grace à une barre d'outils adaptée d'exécuter les algorithmes développés par le groupe 2 et d'afficher les résultats.

L'onglet exécution se présente comme à la FIG. 7 (page 6)

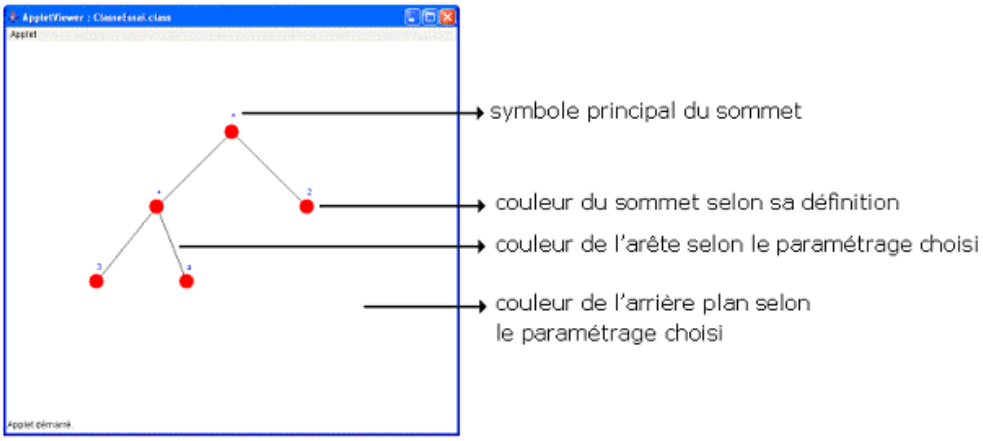


FIG. 4 – Représentation d'un arbre (1)

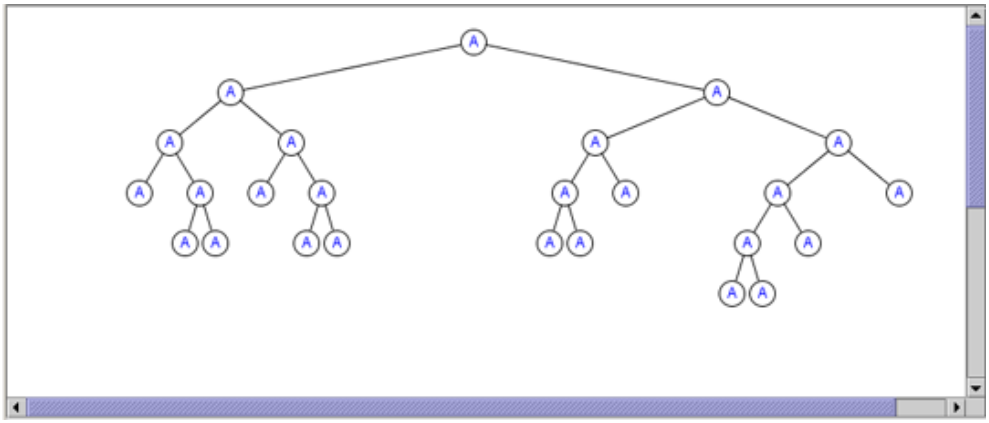

FIG. 5 – Représentation d'un arbre (2)

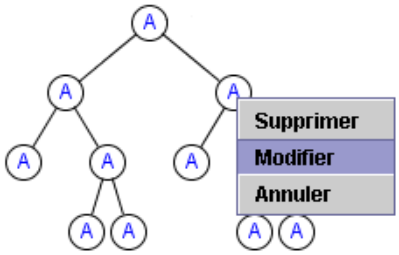


FIG. 6 – Modifier ou supprimer un sommet

- **Barre d'outils**
  Les fonctionnalités proposées par le module "exécution" sont spécifiées dans la barre d'outils, on distinguera notamment :
  - L'édition d'un arbre résultant d'un algorithme.
  - Le lancement/arrêt d'un algorithme sélectionné depuis une liste.
  - L'exécution pas à pas d'un algorithme.

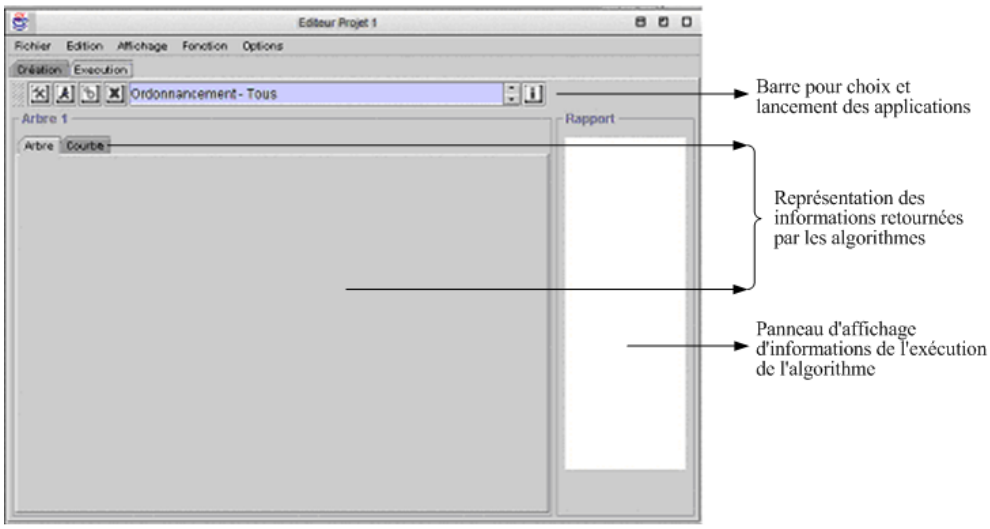


FIG. 7 – Aperçu de l'onglet exécution

L'appel aux méthodes des algorithmes sera fera à partir soit d'une instance d'un module "Algorithme", soit en utilisant des méthodes static de classes abstraites.

- **Représentation du résultat des algorithmes**
  Selon l'algorithme appelé, les informations retournées différeront. On pourra récupérer l'arbre en entrée, sur lequel on affichera des données spécifiques (nombre de Schrall, nombre de registres utilisés...), une liste d'ordonnancement à partir de laquelle on pourra tracer une courbe. Ainsi cette partie est composée de plusieurs composants graphiques dépendants des algorithmes proposés par le groupe 2.

  – Aspect évènements
    A partir d'un arbre, on pourra suivre un sommet en particulier et voir son évolution, afficher des informations relatives à ce sommet (comme stipulé dans le sujet, chaque sommet peut contenir un nombre quelconques d'informations numériques ou textuelles) ou bien afficher des informations relatives aux arêtes, ceci par simple clique de souris.
    Ces évènements sont proposés par un simple clique droit (comme dans la création) sur un sommet (ou une arête), donnant accès au menu contextuel de la FIG. 8 (page 6) :

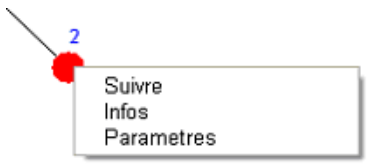


FIG. 8 – Exemple de menu contextuel dans l'onglet exécution

  – Représentation d'une courbe
    La présence d'une courbe traçant l'évolution de l'utilisation des registres a été indiquée comme utile à l'IHM, donc son intégration se fera dans un onglet à part et réalisé à partir d'une liste d'ordonnancement.

### 2.2.3 L'onglet Paramètrage

Le but de cet onglet dans le logiciel sera de proposer à l'utilisateur toute une série de fonctionnalités qui vont lui permettre de personnaliser l'arbre binaire sur lequel il est en train de travailler. Ces fonctions seront divisées en trois parties : une première pour la coloration des différents types de sommets et des arêtes, une seconde pour la coloration des informations contenues dans l'arbre et une dernière pour paramétrer la vitesse d'exécution des threads qui serviront au déroulement pas à pas de certains algorithmes. La Fig. 9 (page 7) montre comment ces différents points pourront s'organiser graphiquement.

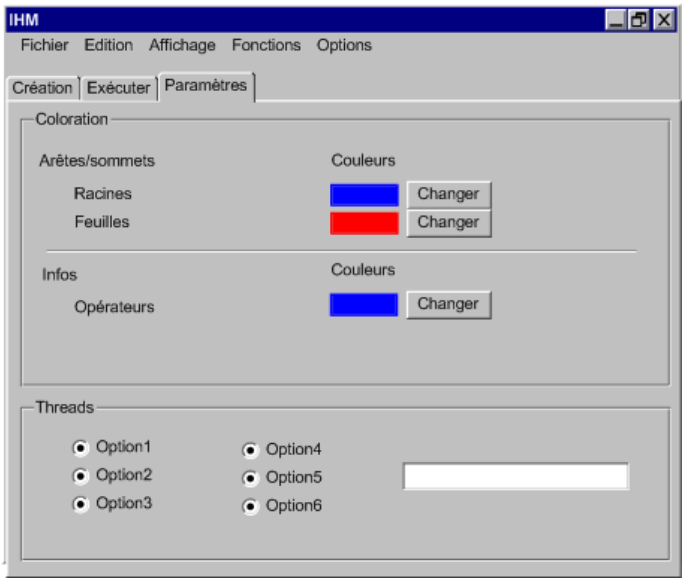


Fig. 9 – Onglet Paramètrage

- **Coloration des sommets et des arêtes**
  Les sommets d'un arbre peuvent être de plusieurs types : racine, feuille, autres sommets. De plus il peut être utile de suivre un sommet lors du déroulement d'un algorithme, c'est pour cela que nous définissons une quatrième catégorie qui s'appellera sommet suivi. L'utilisateur aura donc la possibilité de choisir une couleur spécifique pour chacun des types de sommet sachant que pour suivre un sommet il faut au préalable le sélectionner dans l'onglet exécution (2.2.2 de la page 4). Il sera aussi possible de choisir la couleur de toutes les arêtes de l'arbre pour en avoir une meilleur lecture.
- **Coloration des informations**
  Les arbres binaires que nous traitons vont permettre de manipuler des expressions arithmétiques. C'est pour cela que nous pourrons choisir deux couleurs différentes, l'une pour tous les opérateurs ($+$, $-$, $*$, $/$), l'autre pour les nombres. Ces arbres contiendront aussi un certain nombre d'informations qui pourront être stockées soit dans les sommets soit dans les arêtes. Là encore, deux couleurs pourront être sélectionnées pour faire la différence entre les arêtes et les sommets (en effet ils ne contiendront peut-être pas le même type d'informations). Enfin, si l'utilisateur clique sur un sommet dans l'onglet exécution, il pourra afficher toutes les informations de celui-ci, qui elles aussi, pourront avoir leurs couleurs spécifiques. Enfin, la couleur de l'arrière plan pourra aussi être choisie.
- **Vitesse des threads**
  Pour que l'utilisateur ait le temps de voir correctement le déroulement d'un algorithme ou qu'il puisse utiliser la barre d'outils de l'onglet exécution (par exemple l'option pause), il ne faut pas que le temps entre deux affichages successifs soit trop rapide. Il y aura donc

un espace pour que l'utilisateur entre la vitesse souhaitée. Par exemple : 1s, 2s, 5s, et 10s seront automatiquement proposées mais il pourra tout de même choisir 0,5s, par exemple s'il veut un traitement plus précis.

- **Paramètrage par défaut**

  Si l'utilisateur ne modifie pas les paramètres d'affichage mais qu'il fait quand même afficher les informations d'une arête ou qu'il sélectionne un sommet pour le suivre, les couleurs prédéfinies seront alors utilisées. Elles seront bien sûr choisies de manière à afficher le plus clairement possible les informations demandées.

# Table des figures


1 Diagramme de classes simplifié . . . . . 3
2 Aperçu de l'interface graphique . . . . . 4
3 Intéractions entre les différentes entités du projet . . . . . 4
4 Représentation d'un arbre (1) . . . . . 5
5 Représentation d'un arbre (2) . . . . . 5
6 Modifier ou supprimer un sommet . . . . . 5
7 Aperçu de l'onglet exécution . . . . . 6
8 Exemple de menu contextuel dans l'onglet exécution . . . . . 6
9 Onglet Paramètrage . . . . . 7