IRI
Institut de Recherche Interdisciplinaire

cnrs
dépasser les frontières

Développement multi-sites d'une application graphique J2SE pour l'analyse de données de biopuces « ace.map »

Guillaume Brysbaert

Ingénieur de recherche en bioinformatique

Journée Min2RIEN - 06/12/2012

I. Parcours

Formation initiale

2003 ***Ingénieur en systèmes d'information (ISEN)***

2004 ***DESS de bioinformatique*** à l'université de Lille 1

Expérience professionnelle

I. 2004/2005 – Ingénieur à l'IPL (INSERM U744) : Perl

II. Depuis 2005 – Ingénieur de recherche à l'IRI (USR 3078) : Java

- De 2005 à 2010 : membre de l'équipe « Systèmes épigénomiques » dirigée par Arndt Benecke

  => Développement d'« ace.map » pour l'analyse de données de biopuces

- Depuis 2010 : membre de l'équipe « Nanosystèmes biologiques » dirigée par Ralf Blossey

  => Etude des complexes de remodelage de la chromatine

I. Parcours

Formation initiale

2003 ***Ingénieur en systèmes d'information (ISEN)***

2004 ***DESS de bioinformatique*** à l'université de Lille 1

Expérience professionnelle

I. 2004/2005 – Ingénieur à l'IPL (INSERM U744) : Perl

II. Depuis 2005 – Ingénieur de recherche à l'IRI (USR 3078) : Java

- De 2005 à 2010 : membre de l'équipe « Systèmes épigénomiques » dirigée par Arndt Benecke

  => Développement d'« ace.map » pour l'analyse de données de biopuces

- Depuis 2010 : membre de l'équipe « Nanosystèmes biologiques » dirigée par Ralf Blossey

  => Etude des complexes de remodelage de la chromatine

II. Laboratoire/Equipe

Institut de Recherche Interdisciplinaire

Approche interdisciplinaire de l'étude des réseaux de régulation cellulaire : biologie, chimie, physique, mathématiques, informatique

Equipe « Systèmes épigénomiques » IRI - IHES

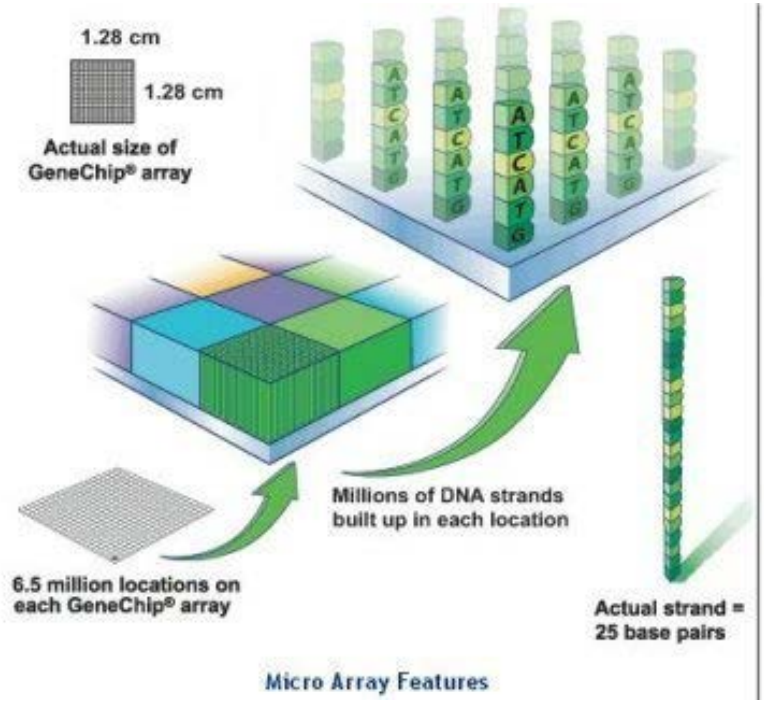
- Etude du transcriptome à l'aide de puces à ADN (niveau d'expression des gènes)
- Expériences réalisées au sein même de l'équipe ou par des collaborateurs
- Analyse des données grâce à « ace.map », logiciel en J2SE développé au sein de l'équipe
- Equipe interdisciplinaire : biologistes, biochimistes, informaticiens, mathématiciens, physiciens
- Répartition sur 2 sites
  - IRI (Lille) : biologistes/biochimistes + moi
  - IHES (Paris) : mathématiciens/physiciens/informaticiens
- Mon rôle : responsable des développements

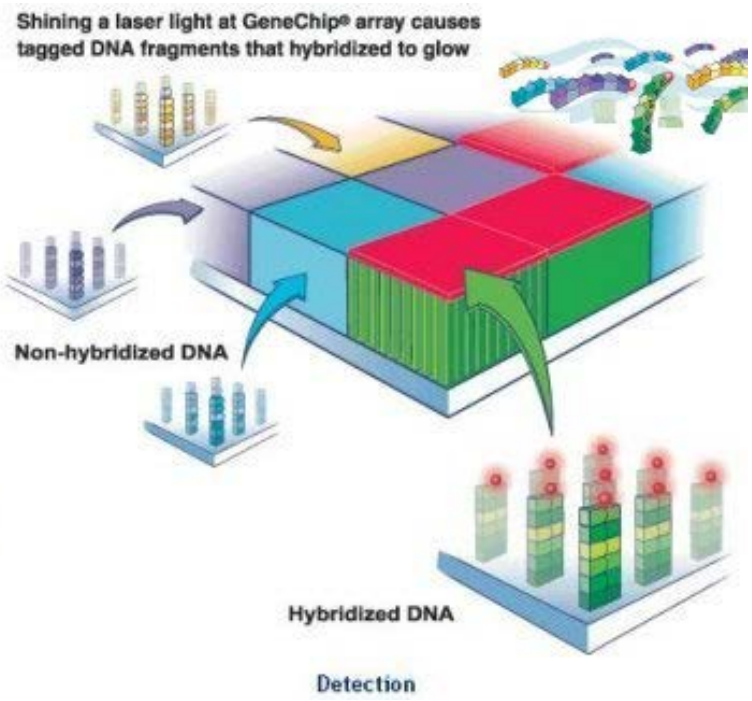
Puces à ADN

But : permet de mesurer le niveau d'expression des gènes

Moyen :

- Mesure de la quantité d'ARN messager dans une cellule.
- Utilisation de sondes (au moins une par ARNm)
- Détection de la luminescence

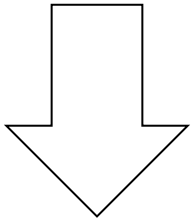




IRI
Institut de Recherche Interdisciplinaire

Résultats

	A	B	C	D
1	Probe_ID	Gene_ID	Assay_Norm	CV_HA00OH
2	100002	hCG1643199.	110.74	0.09
3	100003	hCG2041918	0.33	0.50
4	100027	hCG31426.2	0.42	0.28
5	100036	hCG1979099.	0.51	1.83
6	100037	hCG42687.4	11.08	0.09
7	100039	hCG2015782	28.10	0.04
8	100043	R26_hCT1644	0.30	0.57



Analyse => « ace.map », application stand-alone en J2SE

~55000 lignes de code

2 sites

6 développeurs

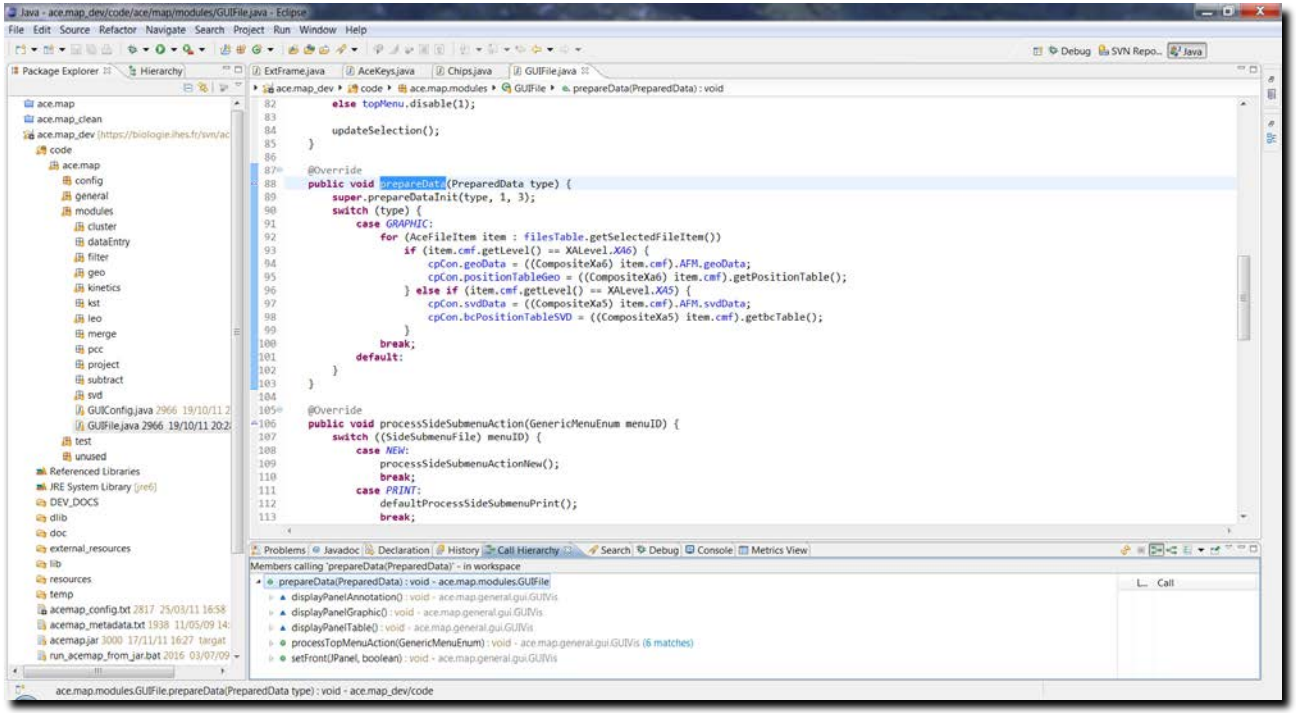
IRI
Institut de Recherche Interdisciplinaire

Environnement de développement : Eclipse

⇒ Eclipse

Avantages

- Pratique pour le développement Java (ergonomie, debug, changement de noms, utilisation de méthodes et classes…)
- Gratuit
- Nombreux plug-in



Inconvénients

- Certains plug-in ne fonctionnent plus très bien avec les nouvelles versions

IV. Outils utilisées

2 sites, 6 développeurs : SVN

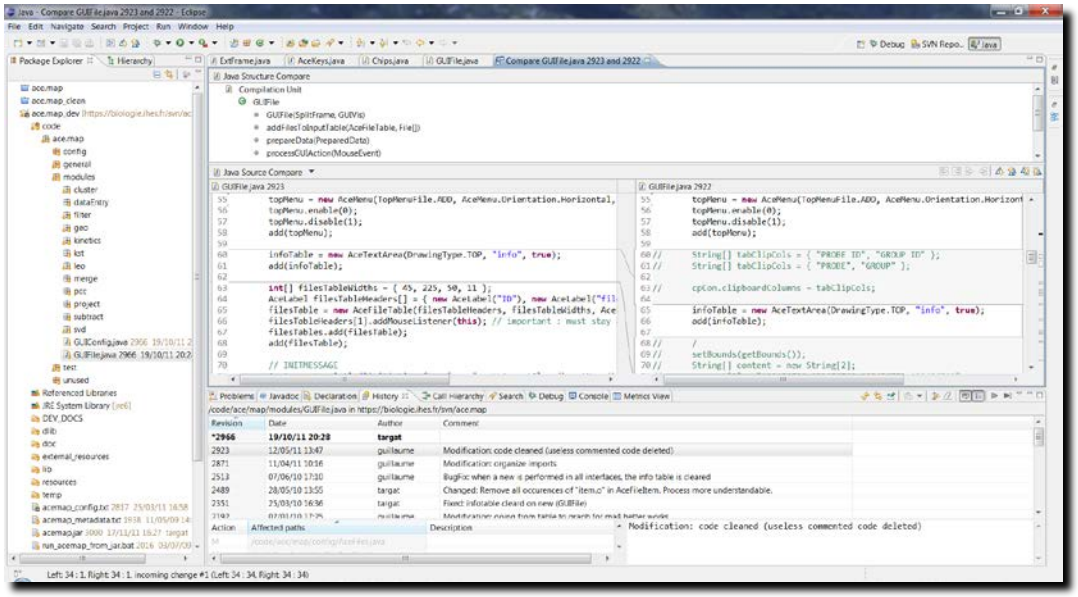
⇒ SVN pour la gestion de version et plug-in Subclipse pour Eclipse

Avantages

- Indispensable pour un développement collaboratif sur le même code source
- Très pratique pour le suivi des développements et le debugging : historique et descriptif des modifications
- Intégration dans Eclipse grâce au plug-in Subclipse

Inconvénients

- Projet grossissant sur le serveur



IV. Outils utilisées

Gestion des bugs : TRAC

⇒ Bugtracking géré avec Trac

Avantages

- Report de bugs par les développeurs et les utilisateurs
- Assignation à une personne et définition de priorités
- Pratique en gestion de projets

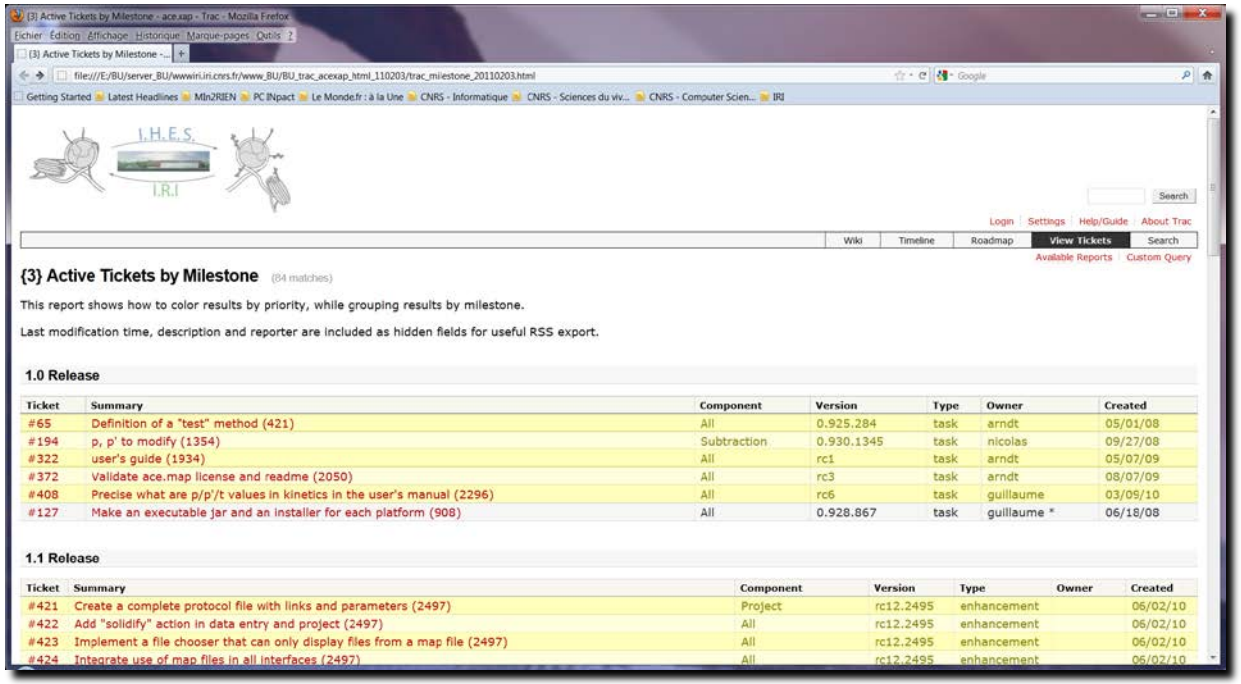
Inconvénients

- Suivi important :

  doit être mis à jour

  régulièrement par

  tous les développeurs



IV. Outils utilisées

Communication : Adobe ConnectNow

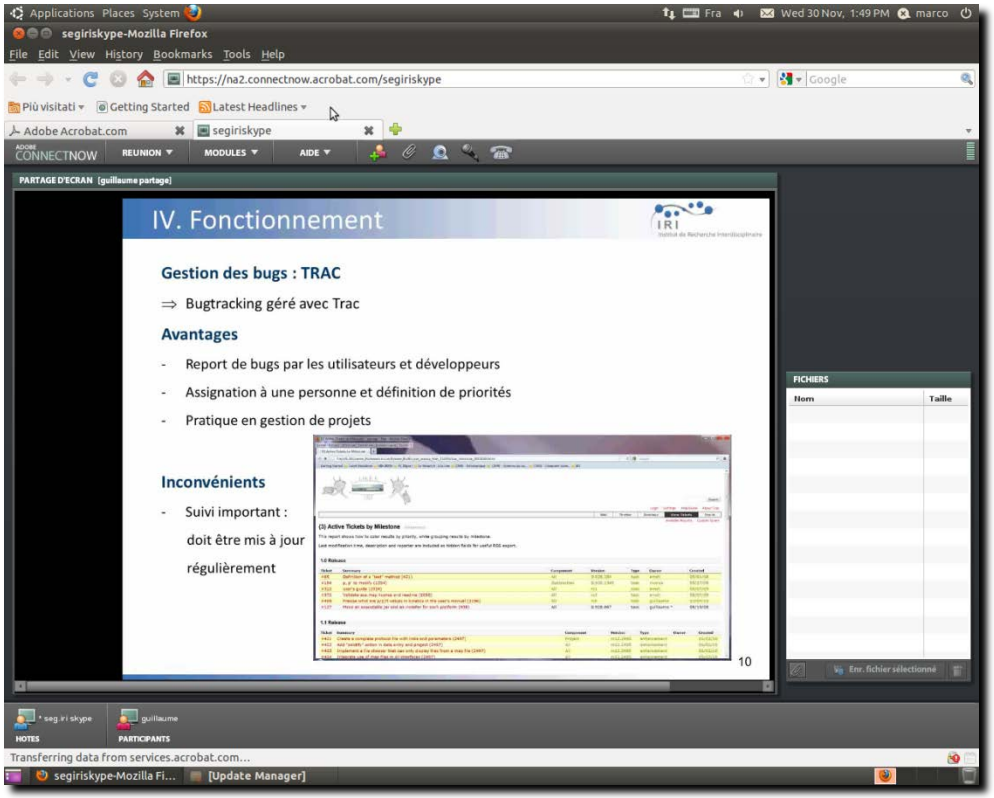
⇒ Partage d'écran géré avec ConnectNow

Avantages

- Possibilité de partager des présentations ou d'afficher du code (Eclipse) pour debugging…
- Gratuit jusqu'à 3 utilisateurs

Inconvénients

- Qualité parfois un peu moyenne



IV. Outils utilisées

IRI
Institut de Recherche Interdisciplinaire

Langage orienté objet : J2SE

⇒ Langage de développement : J2SE

Avantages

- Multi plate-formes
- Pratique pour le développeur (eclipse, javadoc, beaucoup de codes et bibliothèques disponibles sur le web, grande communauté, outils d'intégration continue comme CruiseControl ou Jenkins…)

Inconvénients

- Gestion de la mémoire
- Calculs sur flottants (float)
- Interfaces en SWING/AWT

IRI
Institut de Recherche Interdisciplinaire

Mémoire

- Gérée par le Garbage collector

⇒ Pas d'allocation/désallocation manuelle, uniquement automatique

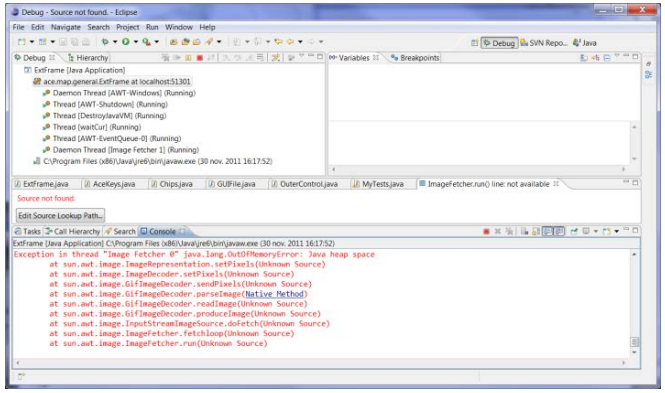
⇒ Pratique pour le développeur

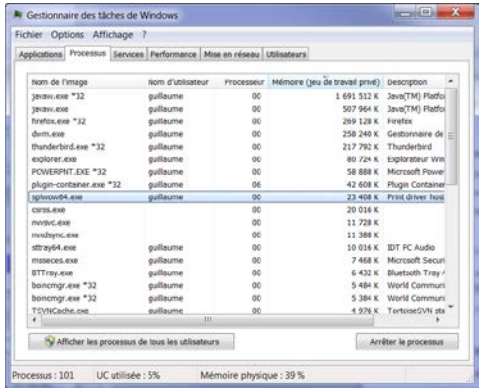
⇒ Mais la désallocation intervient de manière « aléatoire » :

Un objet qui n'est plus utilisé dans le code ne sera pas nécessairement détruit immédiatement

- Limitation en mémoire dans l'exécution de l'application sur la Java Virtual Machine

⇒ Maximum ~1,5 Go de mémoire allouée pour l'application (paramètre –Xmx) –> 32 bits

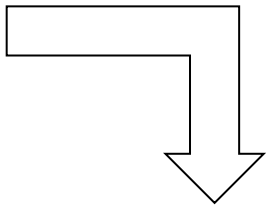




IV. Outils utilisées


```java
// generate a random table of 30000 floating numbers
float[] floatTable = new float[30000];
double[] doubleTable = new double[30000];
Random r = new Random();

for (int i = 0; i < floatTable.length; i++)
    floatTable[i] = r.nextInt() + r.nextFloat();

// create a double table of the same elements
for (int i = 0; i < doubleTable.length; i++)
    doubleTable[i] = floatTable[i];

// float
float sumFloatNoSorted = 0.0f;
float sumFloatSorted = 0.0f;
for (float floatElt : floatTable)
    sumFloatNoSorted += floatElt;
Arrays.sort(floatTable);
for (float floatElt : floatTable)
    sumFloatSorted += floatElt;

System.out.println("SumFloatNoSorted   : " + sumFloatNoSorted);
System.out.println("SumFloatSorted     : " + sumFloatSorted);

// double
double sumDoubleNoSorted = 0.0f;
double sumDoubleSorted = 0.0f;
for (double doubleElt : doubleTable)
    sumDoubleNoSorted += doubleElt;
Arrays.sort(doubleTable);
for (double doubleElt : doubleTable)
    sumDoubleSorted += doubleElt;

System.out.println("SumDoubleNoSorted : " + sumDoubleNoSorted);
System.out.println("SumDoubleSorted   : " + sumDoubleSorted);
```

Calculs sur flottants (float)



```
SumFloatNoSorted   : -7.7484384E10
SumFloatSorted     : -7.7535543E10
SumDoubleNoSorted : -7.748464319071875E10
SumDoubleSorted   : -7.748464319071875E10
```

IV. Outils utilisées

IRI
Institut de Recherche Interdisciplinaire

Développement d'une interface spécifique

⇒ Nouveaux boutons, curseurs, sliders…

⇒ Coût en temps de développement et de maintenance élevé

IRI
Institut de Recherche Interdisciplinaire

Pour une activité pérenne et un code maintenable

```
private void sortGroups(treatmentGroup gr[], int p[], int Ls, int Rs){
    int Lw = Ls;
    int Rw = Rs;

    treatmentGroup pivot = gr[(Lw+Rw)/2];
    treatmentGroup temp;
    int t;

    while(Lw<=Rw){
        while(gr[Lw].compareTo(pivot,  overTime, chngSubs ) <0){
            Lw++;
        }
        while(pivot.compareTo( gr[Rw], overTime, chngSubs ) <0){
            Rw--;
        }

        if(Lw<Rw){
            temp   = gr[Lw];  t     = p[Lw];
            gr[Lw] = gr[Rw];  p[Lw] = p[Rw];
            gr[Rw] = temp;    p[Rw] = t;
        }
        if(Lw<=Rw){
            Lw++;
            Rw--;
        }
    }

    if(Ls<Rw) sortGroups(gr, p, Ls, Rw);
    if(Lw<Rs) sortGroups(gr, p, Lw, Rs);
}
```

- Documenter son code
  - Plus compréhensible
  - Moins de perte de temps
  - Turnover important en recherche
- Donner des noms de variables/méthodes cohérents
  - Mieux vaut un long nom significatif qu'un court incompréhensible : numberOfColumnsToShift ⇔ nocts
  - Minimum si code peu documenté
- Ne pas hésiter à recoder une classe si
  - Le code est trop compliqué et non maintenable facilement
  - L'intérêt et l'utilisation de cette classe sont bien comprises (cas particuliers inclus)

VI. Conclusion

Eclipse : bon environnement de développement

SVN/Trac/ConnectNow : très pratiques pour la gestion de projet

Java : langage intéressant d'un point de vue du développeur et multi-plate-forme mais

- Attention à bien utiliser des double (ou BigDecimal) et non des float pour les calculs
- Attention à la gestion de mémoire
- Ne pas passer trop de temps sur l'interface

A refaire ?

⇒ Plutôt utiliser des clients riches tels que EclipseRCP

⇒ Peut-être dans un autre langage (C++, C...) pour éviter des problèmes de gestion de mémoire… mais

- fonction des compétences des développeurs
- dépendance de la plate-forme

⇒ Intégration continue : CruiseControl ou Jenkins