

1 TO DO list

- Parsing et AST ;
- Sémantique des termes ;
- Sémantique des valeurs ;
- Sémantique des processeurs ;
- Implémenter un système de canaux ;
- Soit finir les extracteurs, soit utiliser de la composition ;
- Plein d'autres trucs... .

2 DONE

3 Log

Vendredi 31 : Début du projet. On a commencé à implémenter les classes termes, valeurs et processus.

4 Conventions

Convention de nommage :

- nom de classe par une majuscule ;
- Pour les noms composés on utilise des majuscules ;
- commentaires en scala doc ;
- Au mieux, sauf pour les case classes, une classe par fichier ;
- Les attributs, afin de les différencier des variables finissent par _m ;
- Indentation à 4 espaces ;

5 Commentaires

Extractors : Je doute encore afin de savoir si c'est la bonne façon de faire. Cette histoire d'objet au lieu de classe etc... me perturbe. Utiliser de la composition peut s'avérer pas si mal. Je ne sais pas ce que tu en penses...

Je sais pas si c'est la bonne façon, mais en tout cas, c'est la manière conseillée par le compilateur plutôt que d'imbriquer des case classes. J'ai testé la composition, ça me paraît un peu moins flexible, et un peu plus lourd : il faut d'abord déclarer un objet du type voulu (du style ValueInteger) et l'envoyer après en paramètre à une classe Value qui va le stocker. Explique moi cette histoire de définir d'abord une classe puis un objet, c'est complètement louche ton truc

Commentaires : Le Java est un langage assez normalisé, beaucoup de *bureaucratie* comme dirait Julius. Pour ce projet, je pense qu'on peut en profiter pour s'habituer aux conventions. Comme doxygene en C++, il existe des générateurs de doc pour Java, et donc Scala. Pour cela, il faut utiliser des commentaires un peu spéciaux. Je regarde sur internet en ce moment même pour en savoir un peu plus. Tu m'expliqueras comment on fait, que je normalise mes commentaires, ils sont faits un peu n'importe comment là. A part ça, je vais tout de suite renommer mes attributs avec le `_m` à la fin. Tu regardes ici <https://wiki.scala-lang.org/display/SW/Writing+Documentation>, j'ai commencé à faire un truc, on va voir si l'algorithme de merge fonctionne bien.

sbt : L'équivalent du Makefile pour Scala c'est l'utilitaire sbt. Pour l'instant faut que tu l'installes :

<http://www.scala-sbt.org/0.12.2/docs/Getting-Started/Setup.html>

Pour compiler, tu te places dans le dossier interpreter. Tu lances sbt. La première fois c'est lon, et ensuite tu exécutes la commande run et c'est parti.