

# Discussion suite à la proposition LEMA

Compte rendu des discussions du 12 avril 2007  
Première réunion EVA-Flo : 11-12 avril 2007

Secrétaire de séance : Nathalie Revol

Cette discussion est née autour de la proposition LEMA présentée par Vincent Lefèvre (cf. ses transparents).

- Matthieu : suggère de mettre aussi les preuves des attributs, par exemple la preuve d'une borne d'erreur.
- Nathalie : quel est l'intérêt de conserver les traces ?
- Vincent : à des fins de vérification, et aussi pour donner des indications sur la façon de procéder si on veut ajouter des propriétés.
- Laurent suggère d'utiliser `svn` pour gérer ces traces.
- Vincent refuse cette idée : `svn` ne permet pas de conserver une information structurée.
- Laurent : comme, par exemple, pour pouvoir appliquer Sterbenz, une information utile est que  $x/2 \leq y \leq 2x$ .
- Vincent : oui, mais dans cet exemple il s'agit de conserver des relations entre valeurs mathématiques, c'est plus difficile entre valeurs calculées.
- Laurent : si on a besoin d'un expert pour la gestion de contraintes entre valeurs, contraintes qui décrivent des polyèdres (comme dans l'exemple), on peut demander à Paul Feautrier.
- Note : comme Paul Feautrier est au LIP, ce n'est effectivement pas compliqué ! Mais souvent les contraintes décrivent des ensembles plus quelconques.
- David demande quel est l'état courant de la proposition.
- Vincent : il s'agit d'une réflexion, avec un début d'écriture de quelque chose de semblable en XSLT. Est également en cours de discussion la comparaison entre OpenMath et MathML.
- David : la description de l'arithmétique sera-t-elle dans le problème ou dans le parser ?
- Vincent va la mettre en MathML, séparément du problème si c'est possible.
- David : quels sont les problèmes que présente Gappa et que cette approche résoudrait ?
- Vincent : cela permettrait d'introduire la précision variable ainsi que les paramètres de façon plus générale.
- Nathalie rappelle le schéma d'EVA-Flo.
- Matthieu : dans ce cas, XML est effectivement bien adapté, pour représenter des obligations de preuve etc. et le programme généré à la fin.
- Laurent approuve.
- Matthieu rappelle qu'un problème de Gappa est qu'il ne prouve pas le code lui-même.