CR de la réunion du 15 décembre 2010

Grigory, Christian H, CAL, Christian A.

* Présentation des transparents de Sébastien à la Software Week Atlas
  + Avec la mention de WAF+SCONS
* Quels axes de réflexions peut-on envisager sur CMT ?
  + Objectifs
    - Répondre à la demande Atlas d’améliorer les performances de l’outil de gestion de conf
    - Faire évoluer CMT de telle façon qu’il exploite plus naturellement les nouvelles technologies de build
      * CMake, WAF,…
      * Portabilité multi-plateforme
      * Interfaçage avec les IDE
      * Parallélisme micro et macro
  + Analyse des bases de CMT :
    - Gestion des paramètres de la configuration
      * Base de données
      * Langage de requêtes
    - Pilotage d’outils de la productivité logicielle
      * Build
      * Génération de doc
      * Tests
      * Distribution
      * …
  + Peut-on découpler les deux volets ?
* Scénarios possibles
  + Interfacer CMT et CMake à travers le mécanisme des fragments
    - Génération basique (« *non-intelligente »*) de CMakefiles
    - Probablement peu de bénéfices immédiats sinon
      * Compréhension de CMake
      * Portabilité, IDE ?
  + Dans un deuxième temps, augmenter la syntaxe CMT (**use**) pour mieux exploiter le parallélisme au niveau projet
  + Ensuite envisager une refonte possible de l’aspect Gestion de Conf pour attaquer le problème dit « AFS » (accès aux nombreux fichiers requirements)
    - Architecture BdD (distribuée ?)
  + Analyse des concepts gérés par CMT
    - Structure package/projet
    - Relation de dépendance
    - Constituants (librairie, applications, documents)
    - Paramètres
    - Actions, Patterns
  + Liaison directe avec les IDE
    - Intégration directe Eclipse, Visual, XCode
    - Méta langage (XML…)