

Elèves-ingénieurs

Jérémy Palmier

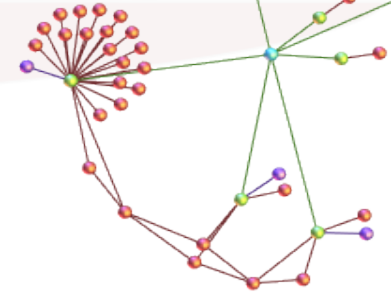
Sébastien Heymann

Microscope numérique de logiciels

"Révéler l'organisation interne de ces systèmes complexes."

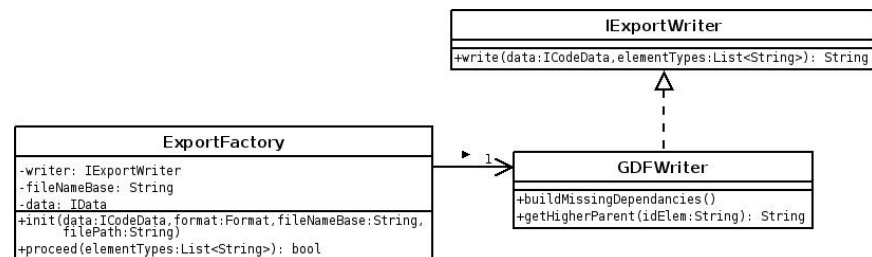
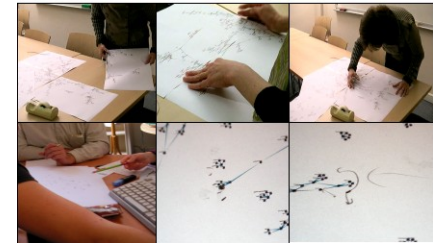
Suiveur UTC

Mr Antoine Jouglet

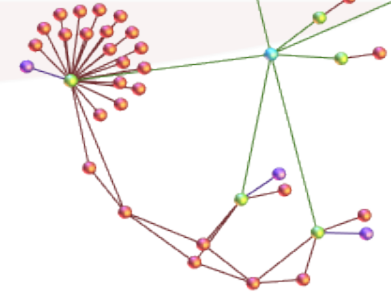


Pourquoi ?

- Approche issue des sciences humaines : infoVis, cartographie de l'information
- Rétro-ingénierie pour la représentation
- Limites de l'UML

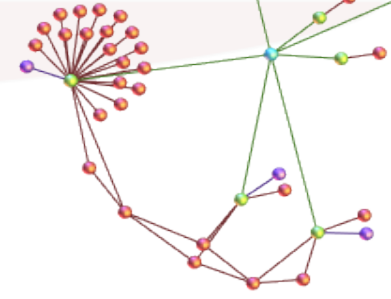


=> Comment comprendre rapidement la nature des interactions entre classes d'un programme ?



Plan

1. Objectifs
2. Technologie et étapes
3. Méthodologie, réalisation et retour sur les usages
4. Projet : gestion et difficultés
5. Bilan et perspective



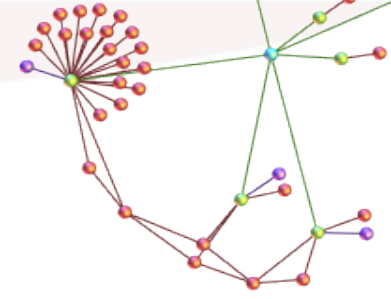
Objectifs

Globaux

- Représenter un logiciel sous forme de graphe et manipuler cette représentation
- Faire émerger des usages

TX

- Synthétiser la structure d'un programme Java et la donner à voir à son développeur ou architecte pour qu'il s'en saisisse

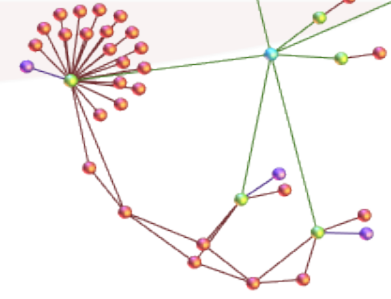


Technologies

- Parseur JDT (Eclipse) étendu à nos besoins
- Logiciel de spatialisation et de manipulation de graphes Gephi (WebAtlas)

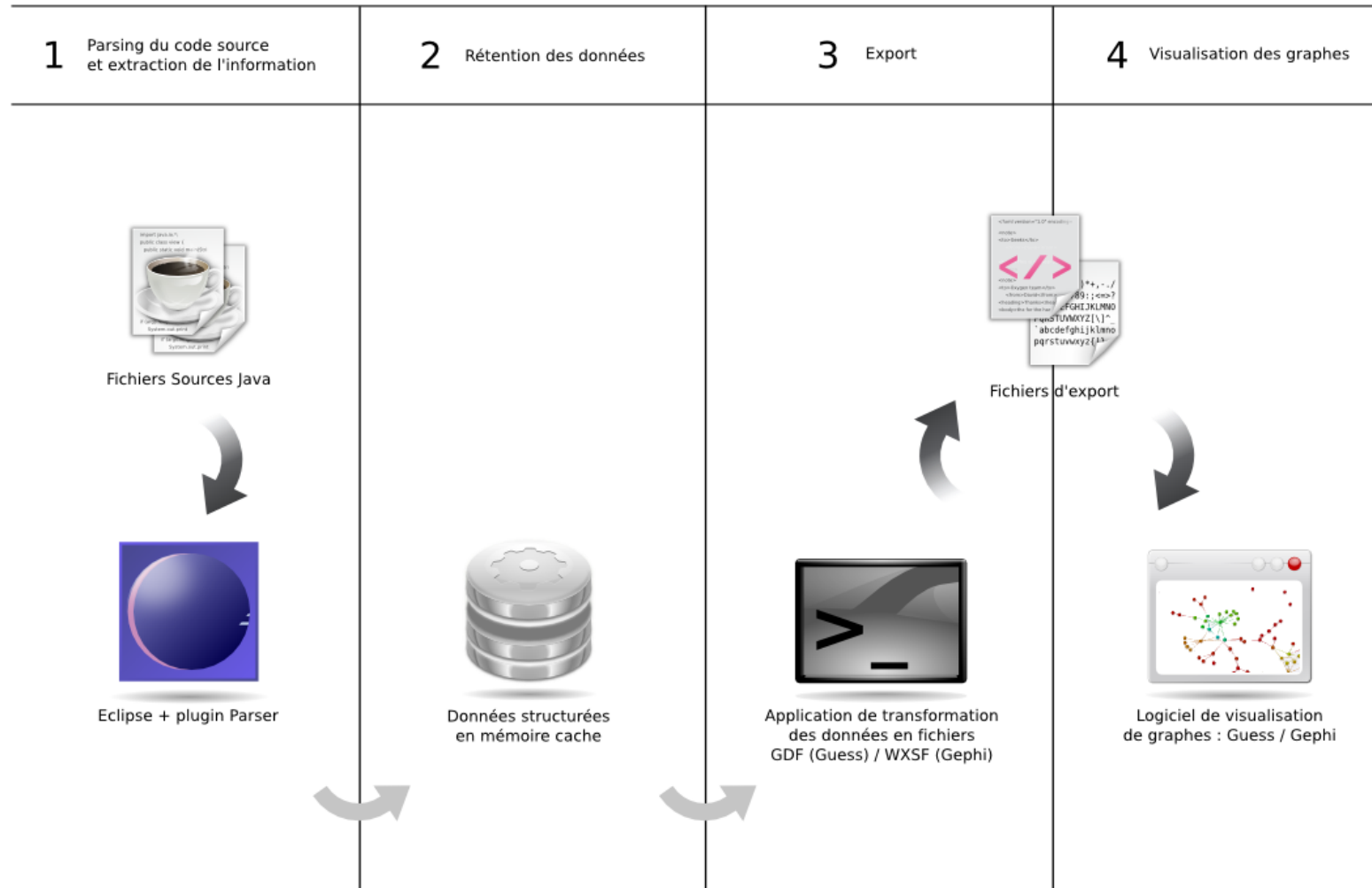


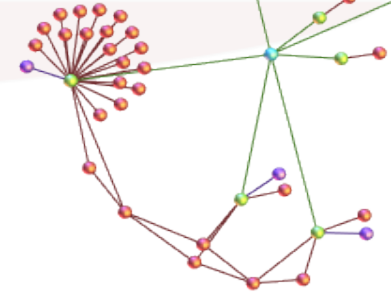
=> Code léger et robuste,
très courte durée de développement (6 semaines)



Etapes

Architecture Overview





Choix de représentation

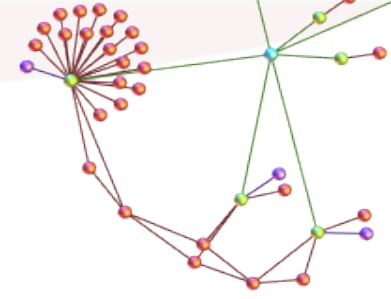
Que représenter dans la structure d'un programme Java (paradigme objet) ?

Éléments

- packages
- classes
- méthodes
- attributs de classe

Relations

- dépendances d'utilisation (appels) entre méthodes
- parenté (arbre AST)
- héritages
- implémentations d'interface



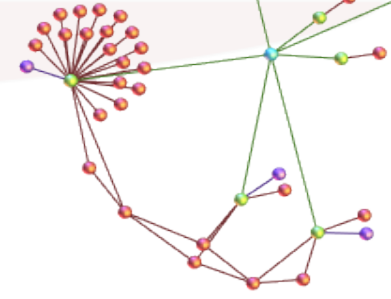
Méthodologie

Génie Logiciel

- Tests unitaires
- Normalisation de la structure de données en 3FN

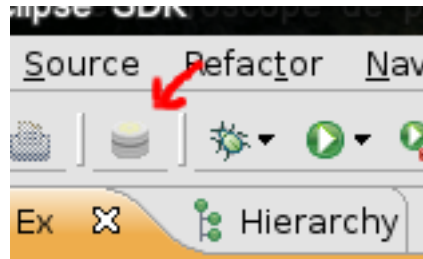
Conception

- Points critiques : création d'algorithmes et calcul de leur complexité
- Méthodologie UML



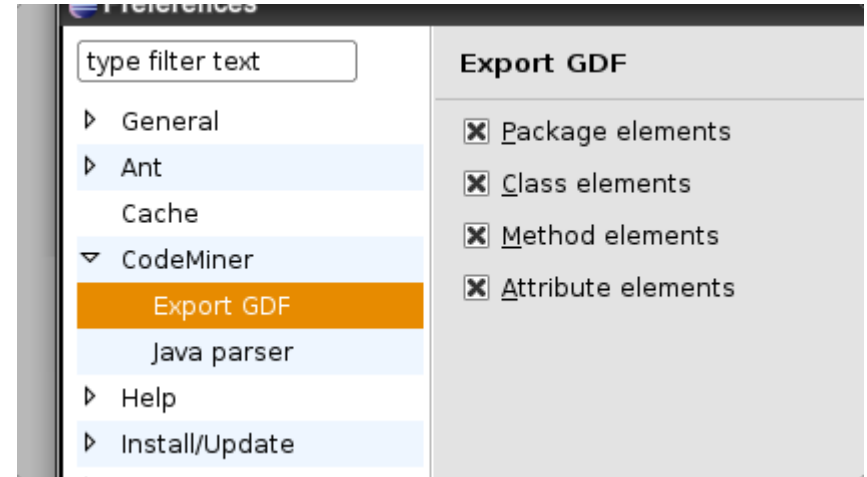
Réalisation : plugin Eclipse

Lancement



icône de démarrage

Préférences



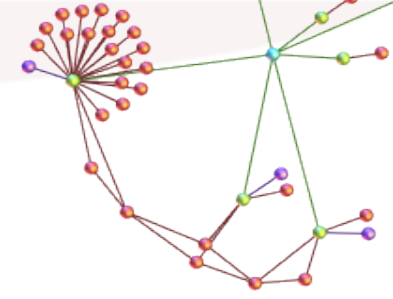
options d'export des données



options générales

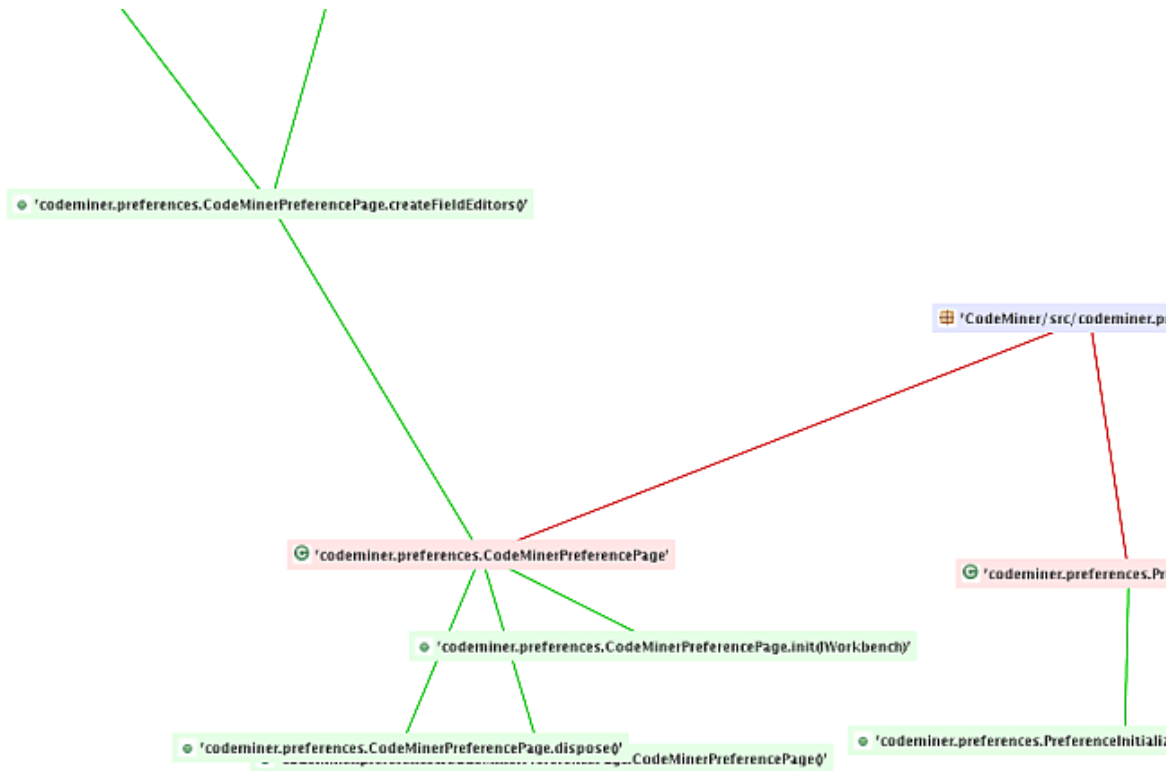


options du parseur



Réalisation : plugin Géphi

47.0



CodeMiner [X]

Eléments courant :
public
codeminer.actions.ExportCommand

Eléments utilisés : 19
Eléments utilisant : 1
Lignes de code : 253

Commentaire

<p> The class ExportCommand implements workbench action delegate. The action is executed in the run function (@see ExportCommand.run()) The run methods would call a parserWebAtlas says THIS is A coma REPLACER if the

Recherche d'éléments

Classe
Export

2 / 5 [Q] [←] [→]

panneau de contrôle (vue détaillée)

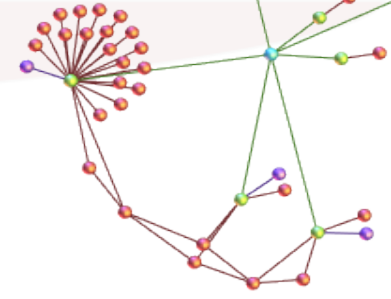
iner/ src/ codeminer.actions'

'CodeMiner/ src/ codeminer.ui'

'c.oueminer/ src/ (default

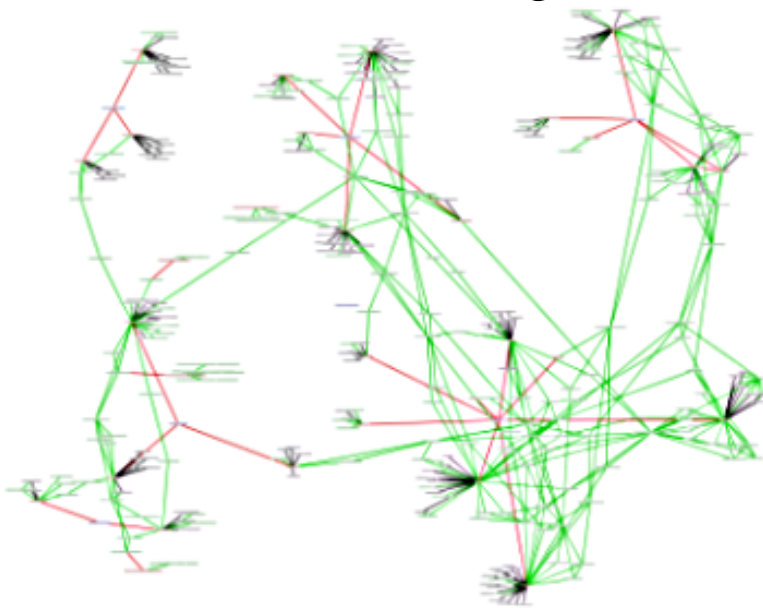


espace de représentation et de manipulation du graphe (vue synoptique)



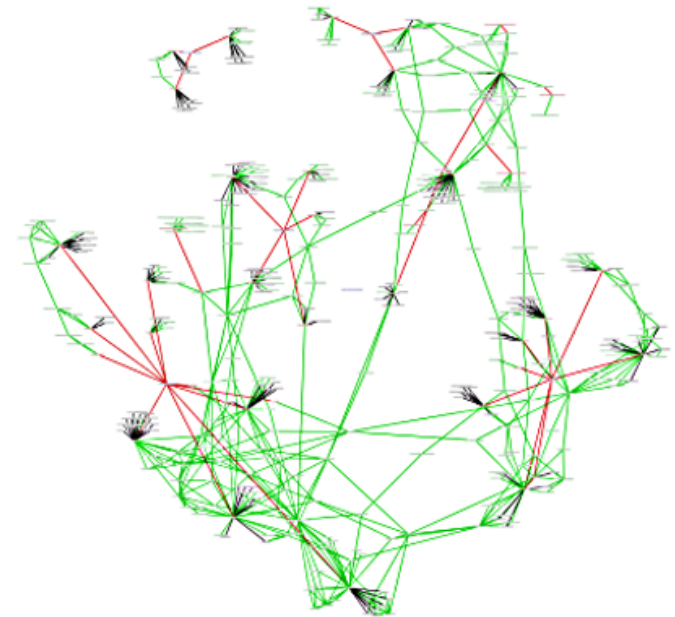
Retour sur les premiers usages

- Mathieu Bastian (Gephi) : validation et orientation dans la structure
- Julian Bilcke (NF28 PhotoViz) : détection de code mort et de travail de binômes mal intégré



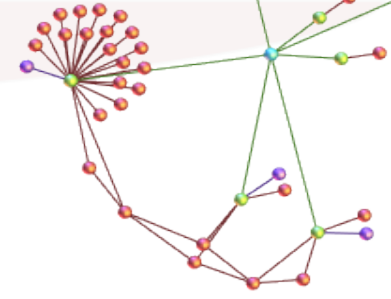
PhotoViz à t=début de développement

« chacun travail dans son coin »



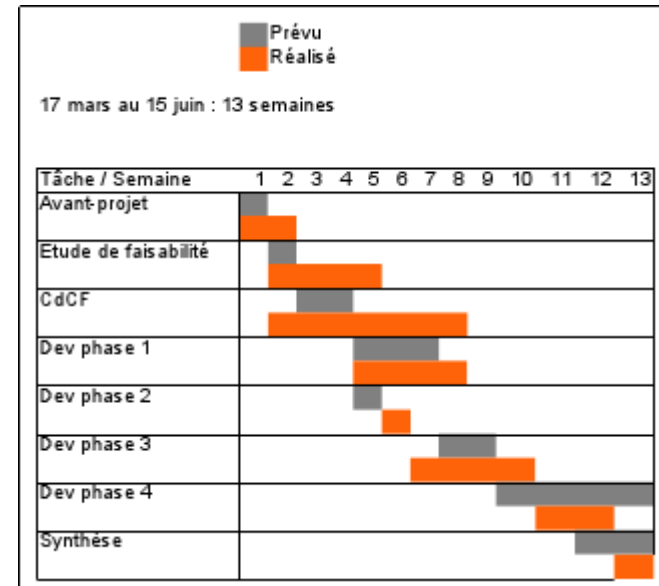
PhotoViz à t + 3 semaines

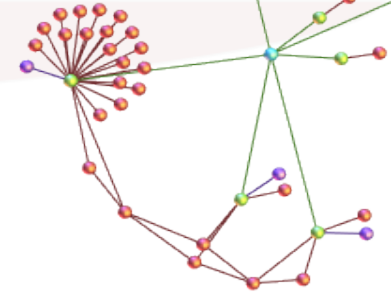
« Intégration des parties »



Gestion du projet

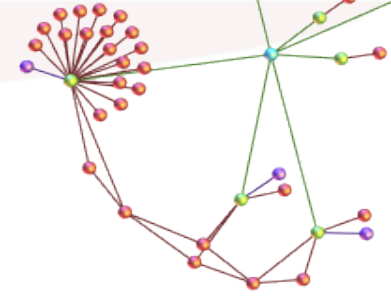
- Respect du planning
- Répartition des tâches selon nos compétences
- Réunions régulières :
réunion projet / semaine,
inter-projets TX (SIGL:CPAN et IceCrawler) / 3 semaines
- Production de rapports (avant-projet, CdCF, final), notes de clarification, cadrage théorique





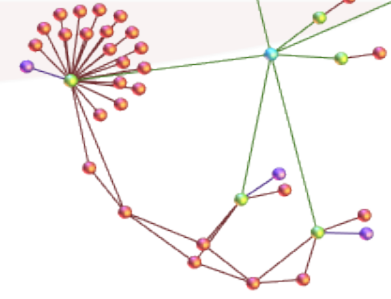
Difficultés surmontées

- Multiples compétences d'ingénieur en R&D
- Délais très courts (3mois)
- Recherche d'informations fiables sur Eclipse et JDT
- Incomplétude de Gephi en version alpha (compensé par la disponibilité du responsable Mathieu Bastian)
- Effectuation des choix technologiques



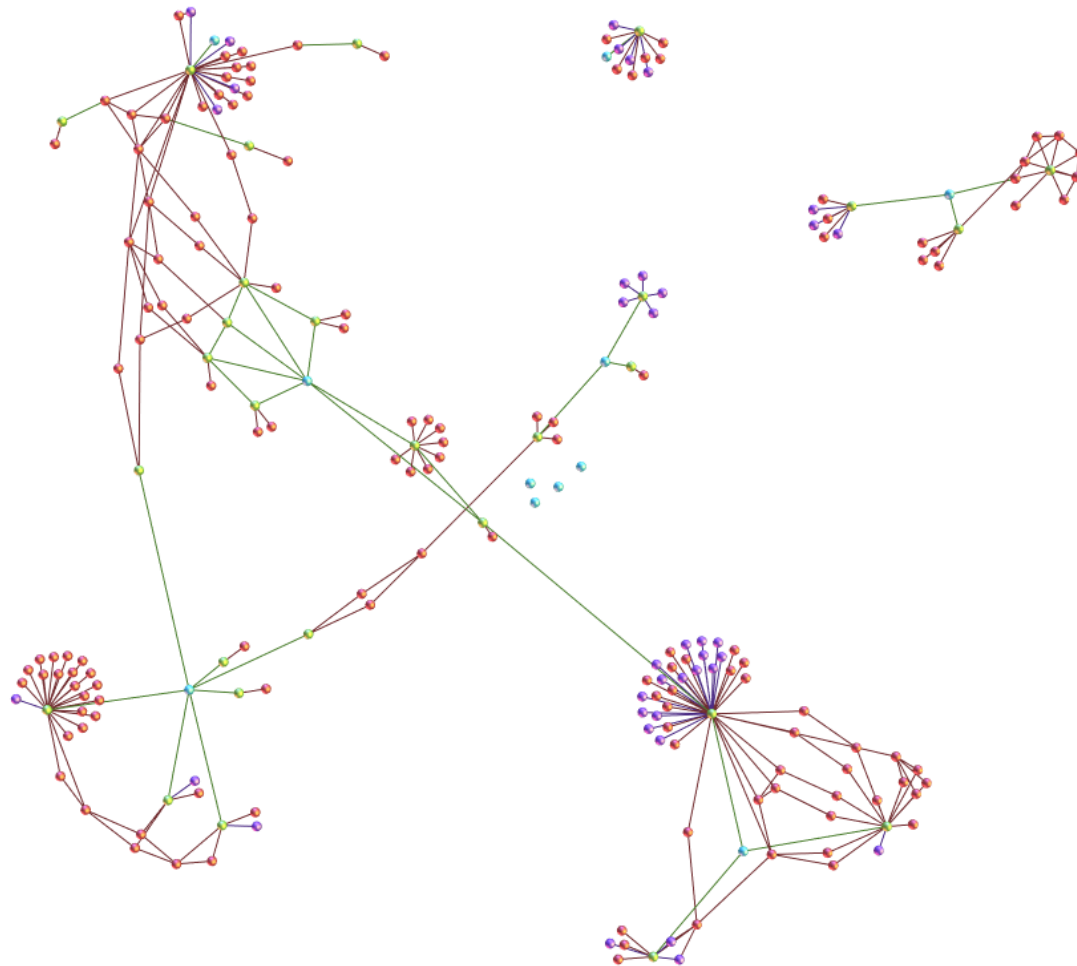
Bilan d'un pari risqué

- ✓ Mise en oeuvre du processus complet d'élaboration d'un nouveau produit
- ✓ Un projet d'innovation robuste et extensible (100% modulaire)
- ✓ Logiciel Libre (GNU/GPL 3) publié sur www.web-mining.fr/codeminer
- ✓ Règle des "3U" atteinte : **Utile, Utilisable, Utilisé !**



Perspectives d'un projet bien né

- Capable de générer des usages
- Amélioration de la visualisation (ex: empreinte de séquences d'instructions)
- Ajout d'autres langages : C++...
- Ajout de fonctionnalités (ex: recherche de cycles)
- Comparaison de graphes dans le temps par analyse de dépôt SVN



Elèves-ingénieurs

Jérémy Palmier

Sébastien Heymann

Microscope numérique de logiciels

"Révéler l'organisation interne de ces systèmes complexes."

Suiveur UTC

Mr Antoine Jouglet