TANAGRA

Un logiciel open source pour l'enseignement et la recherche

Ricco RAKOTOMALALA
Laboratoire ERIC
Université Lumière Lyon 2

http://chirouble.univ-lyon2.fr/~ricco/tanagra/

PLAN

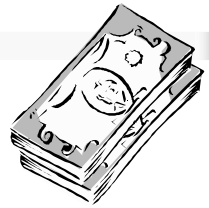
1. Objectifs du projet
2. Le logiciel TANAGRA
3. Distribution et droits
4. Participer au développement
5. Utiliser le logiciel TANAGRA
6. Conclusions et perspectives

1. Objectifs du logiciel TANAGRA

Les logiciels de DATA MINING

Le vrai filon de l'ECD : les logiciels de Data Mining

Essayer une recherche « 'data mining software' sur Google » : # 40,000 références
Sur la page « http://www.kdnuggets.com/software/suites.html » : 7 / 8 commerciaux



L'opposition « commercial » - « recherche »

Interface graphique et fonctionnalités utilisatrices
Mode console et code source libre, installation folklorique

Véhiculer le dynamisme du labo

SIPINA – 695 références sur Goggle
#4 e-mails par semaine à propos de SIPINA
Articles et études en coopération avec d'autres chercheurs

Spécifications du logiciel TANAGRA
A qui s'adresse TANAGRA ?

Un logiciel pour l'enseignement : le profil « chargé d'études »

Les cours, explication des méthodes, outil pédagogique

Les études « réelles » - les « dossiers » - les chercheurs des autres domaines

(cf. tutoriaux études de cas)

Une plate-forme pour la recherche : le chercheur en DATA MINING

Plate-forme d'expérimentation – Tester des méthodes et comparer les résultats

Modularité et accès au code – Programmer ses propres méthodes

(cf. tutoriaux évaluation des méthodes)

Un outil pédagogique pour l'apprentissage de la programmation

Spécifications et conception de ce type de logiciel - Apprendre par l'exemple

Connaître les outils et les bibliothèques types

(cf. page web outils et bibliothèques)

« Open Source » ?

Valider le code = valider les publications

Comparer les résultats
Lecture du code par d'autres chercheurs (ex. du text mining par SD)
Reproduire « exactement » les expérimentations (ex. tirage aléatoire)

Comparer les implémentations

Comparer les interprétations d'un même problème (ex. Bayesien naïf, boosting)
Optimiser le code avec différentes versions

Outil ouvert = Outil vivant

Introduire ses propres algorithmes
Discuter sur la base de prototypes et d'évolutions
Monter et partager des bibliothèques types (ex. générateur de nombres aléatoires, fonctions de répartitions, pourquoi pas des bibliothèques de DATA MINING ?…)

2. Fonctionnalités du logiciel TANAGRA

Fonctionnalités d'un logiciel de DATA MINING

Accès et préparation des données — 1/5
Accéder à un fichier / une BD
Rassembler des sources différentes

Méthodes de Fouille de données — ++/5
Lancer les calculs avec différents algorithmes
Bibliothèque de méthodes

Enchaîner les traitements — 4/5
Faire coopérer les méthodes sans programmer

Évaluer les connaissances — ++/5
Validation croisée, etc.

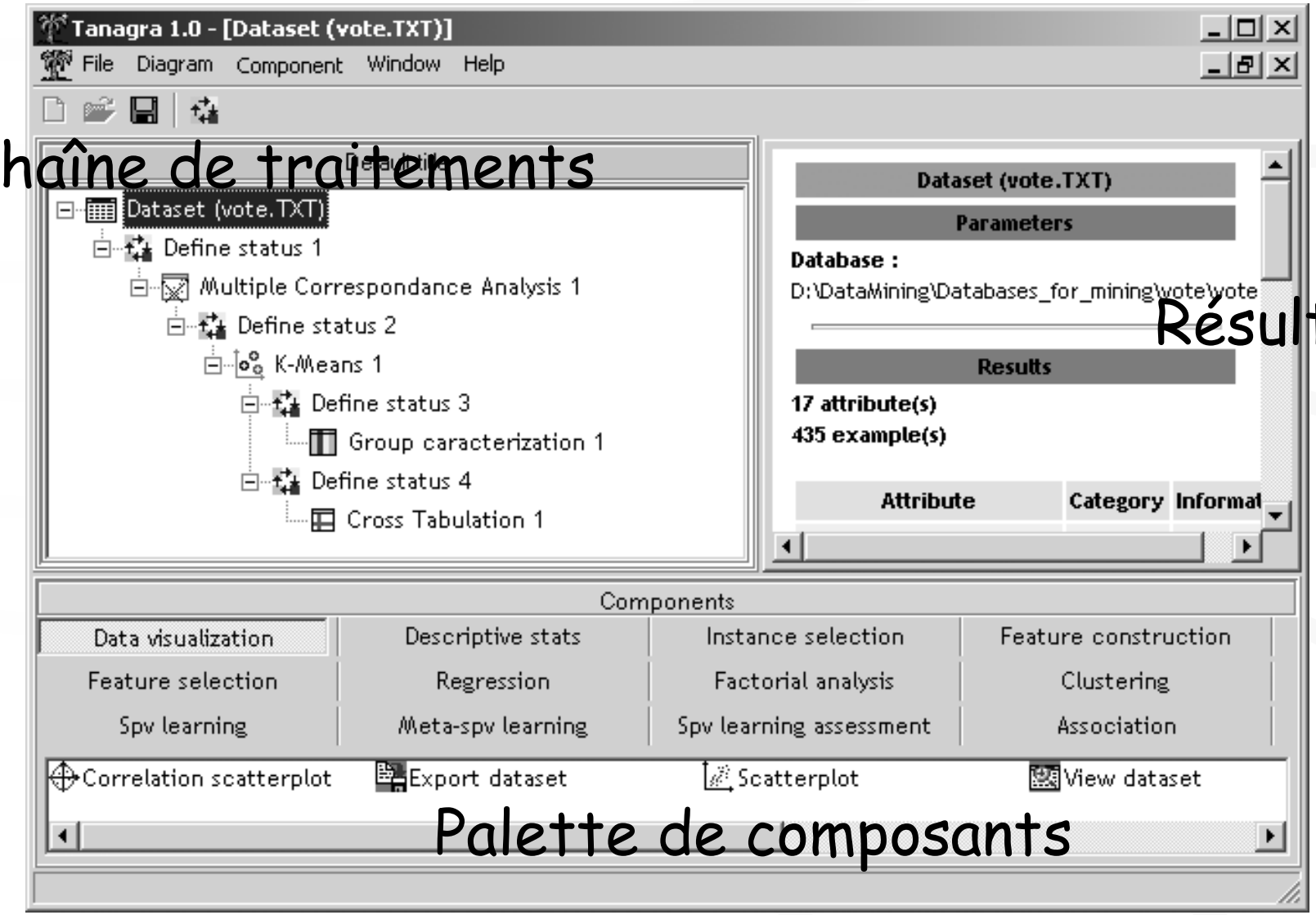
Exploiter les sorties — 2/5
Rapports, visualisation interactive, etc.

Appliquer/exploiter les modèles — 0/5
Modèles en XML, code C, DLL compilées
Prédiction directe sur de nouveaux fichiers

— Logiciels commerciaux
— Prototypes de recherche

TANAGRA

Mode de fonctionnement



Accès aux données

Fichier texte (séparateur tabulation)

Chargement en mémoire
500.000.000 individus théoriques
250.000 individus max pour les règles d'association – EZDL
500.000.000 variables théoriques
Variables continues codées SINGLE
Variables discrètes codées BYTE (255 modalités max)

Quelques éléments sur les performances

COVTYPE – 581.102 ind x 55 var (discrètes) : 240 sec
WAVEFORM – 100.000 ind x 22 var (21 continues) : 20 sec

Formats de sauvegarde

Que sauvegarder ?

La description du traitement - Pas les résultats
Nécessité de ré-exécuter à la prochaine ouverture

Formats

Binaire : intègre les données ⇨ rapidité (covtype = 1,5 sec)
Textuel (fichier INI) : script basique ⇨ souplesse

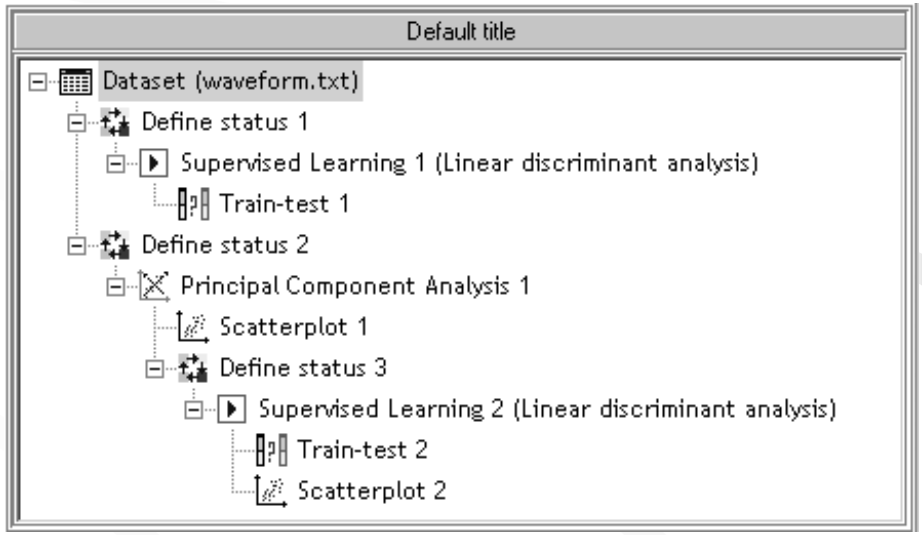
Les méthodes

Les grandes familles aux affaires

Méthodes statistiques
Visualisation

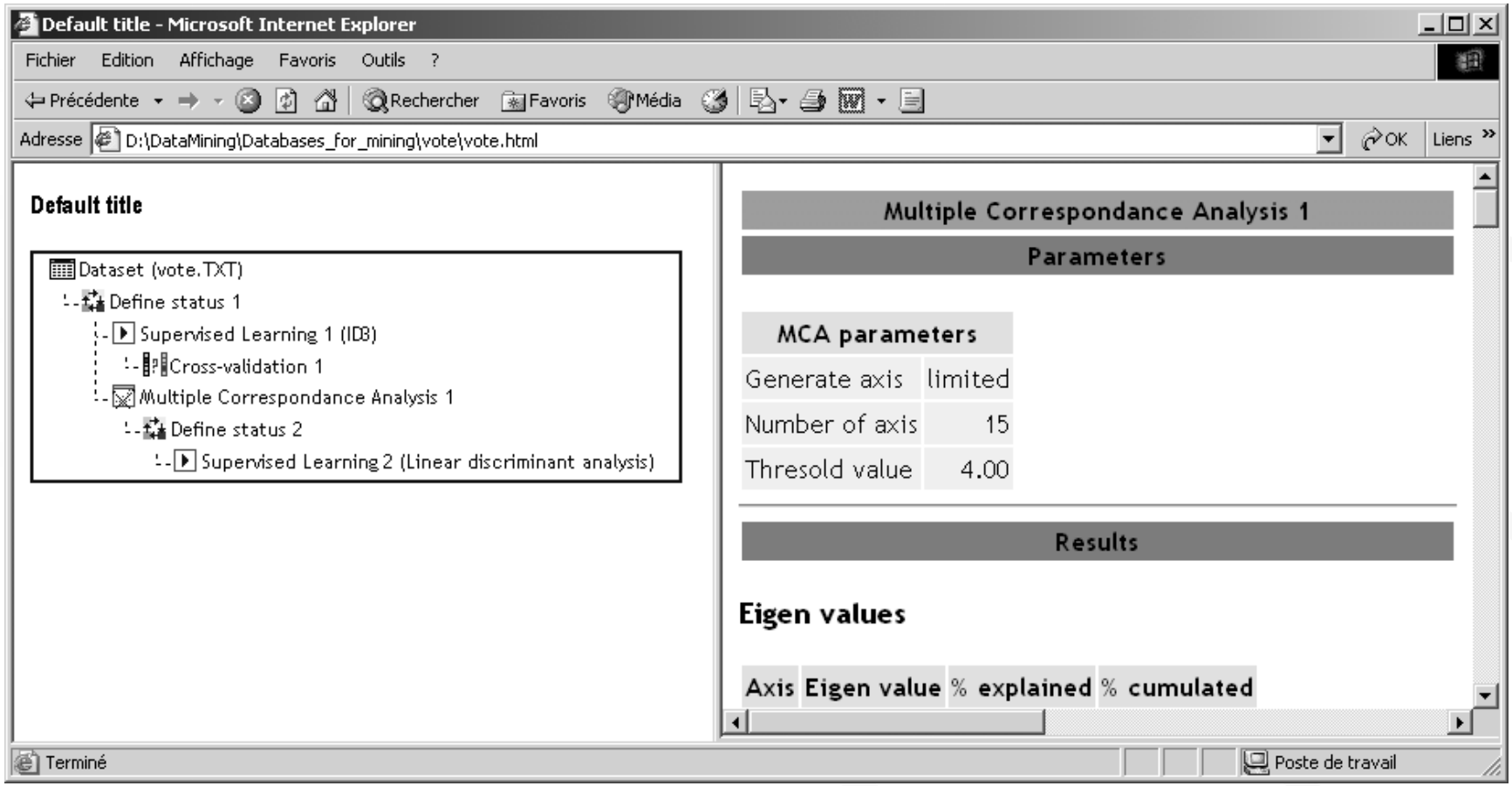
Description – Analyses factorielles
Apprentissage non-supervisé (structuration)
Apprentissage supervisé (prédiction - explication)
Évaluation de l'apprentissage supervisé
Régression
Association

L'enchaînement des méthodes



Les sorties

Privilégier le format HTML
Sortie texte = minimum de code
Formatage HTML reconnu par tous les logiciels
Édition de rapports sans code supplémentaire



3. Accès et licence de distribution

Accès au logiciel

Site
http://chirouble.univ-lyon2.fr/~ricco/tanagra/

Qu'est-ce qui est disponible ?
Setup
Documentation des méthodes et didacticiels
Code source
Documentation du code source
(cf. le site)

Licence

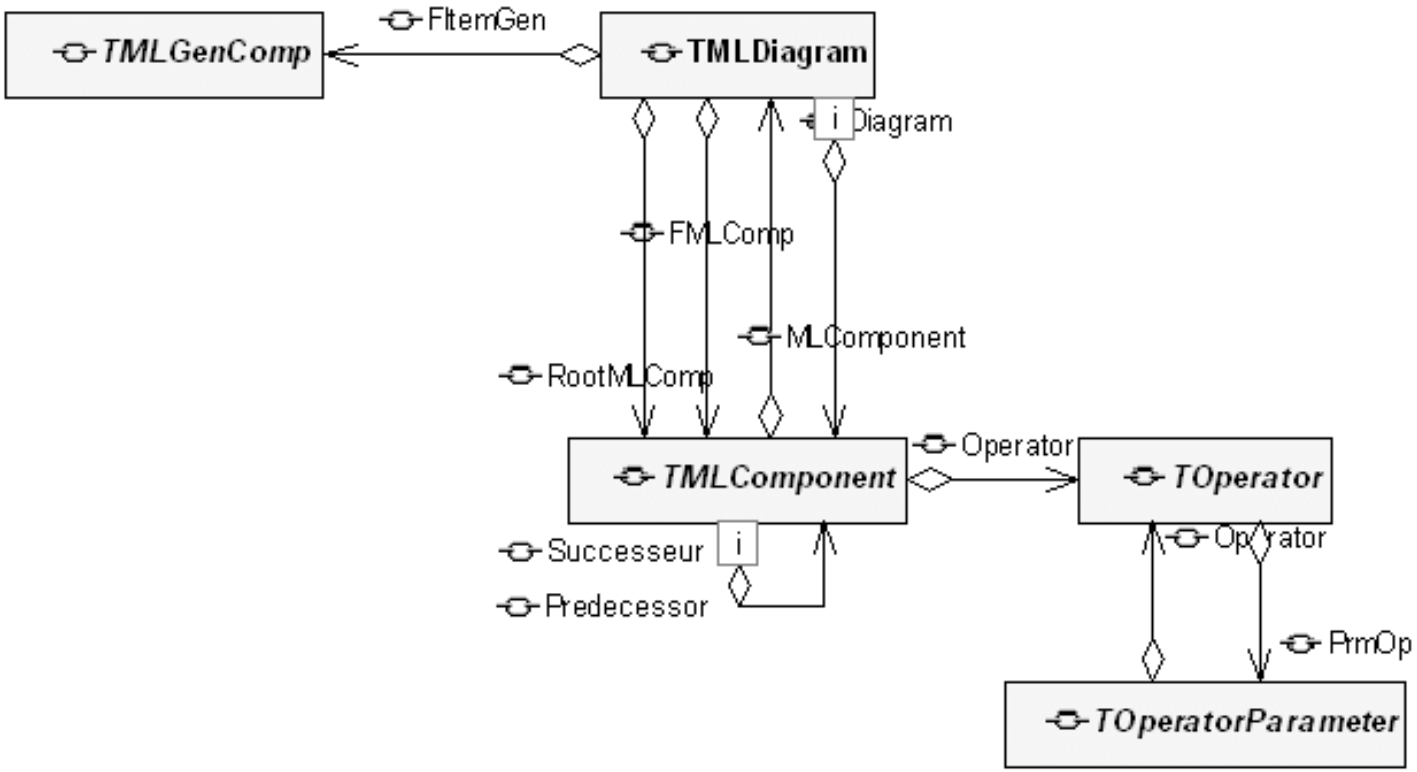
Qui protéger ?
Les utilisateurs : ne pas soustraire un logiciel déjà proposé
Les chercheurs : publier à partir d'un code vérifiable
Les développeurs : garder la propriété de son développement

Comment protéger ?
Inspiration : GPL et OpenSource.org
Principaux points :
- TANAGRA toujours gratuit – Devoir de citation
- Code toujours accessible librement
- Développeur module = propriétaire module
- Module introduit = Module non soustrayable

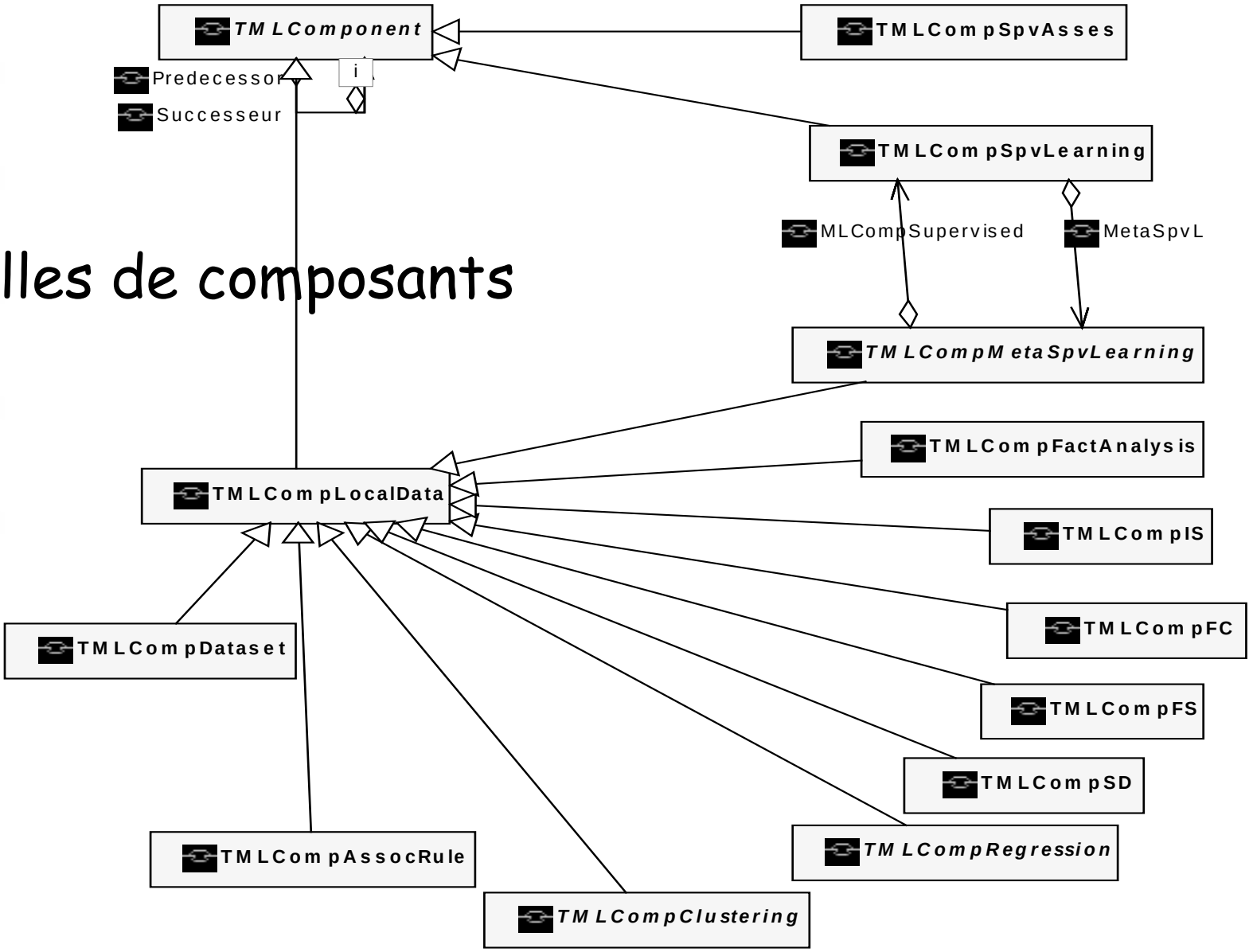
(cf. le fichier de licence)

4. Développer dans TANAGRA

Le modèle d'organisation



Familles de composants



Ajouter un composant

```
<?xml version="1.0" encoding="windows-1252" standalone="yes"?>
<!DOCTYPE components [
  <!ELEMENT components (component+)>

  <!ELEMENT component (name,bitmap,description)>
  <!ATTLIST component class_name ID #REQUIRED>

  <!ELEMENT name (#PCDATA)>
  <!ELEMENT bitmap (#PCDATA)>
  <!ELEMENT description (#PCDATA)>

]><components>

        <component class_name="TMLGenCompViewData">
                <name>
                View dataset
                </name>

                <bitmap>
                MLViewDataset.bmp
                </bitmap>

                <description>
                To visualie current dataset into a grid, values cannot be modified.
                </description>

        </component>
```

Les outils de développement

Points communs ?
(1) Gratuits
(2) Si possible accès au sources
(3) Compatibles KYLIX
(cf. le site)

Type d'outil	Outil	Caractéristiques
Compilateur	Borland Delphi 6	passage KYLIX aisé ?
Bibliothèque de calcul	ATHANOR	Calcul matriciel, optimisation, nombres aléatoires
Bibilothèque de classes	EZDSL	Table de hachage, tableau de bits
Bibliothèque graphique	LMD SE	à remplacer par JEDI
Parser XML	XML Parser	Lecture et validation d'un fichier XML
Visionneuse HTML	HTML Lite	Affichage rapide page WEB (string)

5. Scénarios d'utilisation

Quelques scénarios d'utilisation

1. Données, régression et sorties HTML (autompg)
2. Comparer deux algorithmes supervisés, K-ppv et ID3 (heart)
3. Un exemple de régularisation pour la LDA (wave)
4. Caractérisation d'un clustering (vote)
5. Performances et capacités de calcul (covtype)

6. Conclusions et perspectives

Conclusions

Un support pour les cours
Ne plus dépendre du bon vouloir des dinosaures
Un outil que l'étudiant peut reprendre en stage et en entreprise

Un outil pour les publications à venir
Monter les expérimentations
Discuter des implémentations

Perspectives

Diffuser - documenter
Obtenir le maximum de retour

Déboguer
3-4 mois minimum

Ajouter des fonctionnalités « utilisateurs »
Exécution batch
Format XML du fichier de sauvegarde : script

Ajouter / tester des nouvelles méthodes
Cela dépend de la recherche et des idées