1.Introduction

Travailler avec Sipina sous Linux via Wine.

Je suis dans la période où je (re)découvre Linux. Nous avons vu récemment qu'il était possible de travailler avec Tanagra sous Linux via Wine, simplement, sans contorsions compliquées (http://tutoriels-data-mining.blogspot.com/2009/01/tanagra-sous-linux.html).

Nous montrons dans ce document qu'il est possible de faire de même avec Sipina. Toutes les fonctionnalités du logiciel sont accessibles. On pense notamment aux outils interactifs qui permettent de guider la construction de l'arbre et d'explorer finement les sous-groupes d'observations associées aux nœuds.

Dans ce didacticiel, nous étudierons comment :

1. Installer Sipina sous Linux ;
2. Lancer le logiciel ;
3. Charger un fichier de données au format texte avec séparateur tabulation ;
4. Définir la variable à prédire et les variables prédictives ;
5. Subdiviser aléatoirement les données en échantillon d'apprentissage et de test ;
6. Construire l'arbre sur la partie apprentissage ;
7. Évaluer ses performances sur la partie test ;
8. Explorer finement les sous-ensembles d'observations circonscrites par les nœuds à l'aide des statistiques descriptives comparatives.
9. Initier de nouvelles analyses en prenant comme point de départ un des nœuds de l'arbre.

Nous ne nous étendrons pas outre mesure sur ces fonctionnalités qui sont largement présentées par ailleurs dans plusieurs tutoriels accessibles sur notre site web (http://eric.univ-lyon2.fr/~ricco/sipina.html, voir la section DOWNLOAD ; voir aussi http://sipina.over-blog.fr/). Notre principal objectif dans ce tutoriel est de montrer qu'il est possible d'utiliser Sipina sous Linux.

Nous utilisons la distribution française de Ubuntu 8.10 (http://www.ubuntu-fr.org/). Nous avons également installé WINE, un outil extraordinaire qui permet d'exécuter un très grand nombre de logiciels initialement compilés pour Windows (http://doc.ubuntu-fr.org/wine).

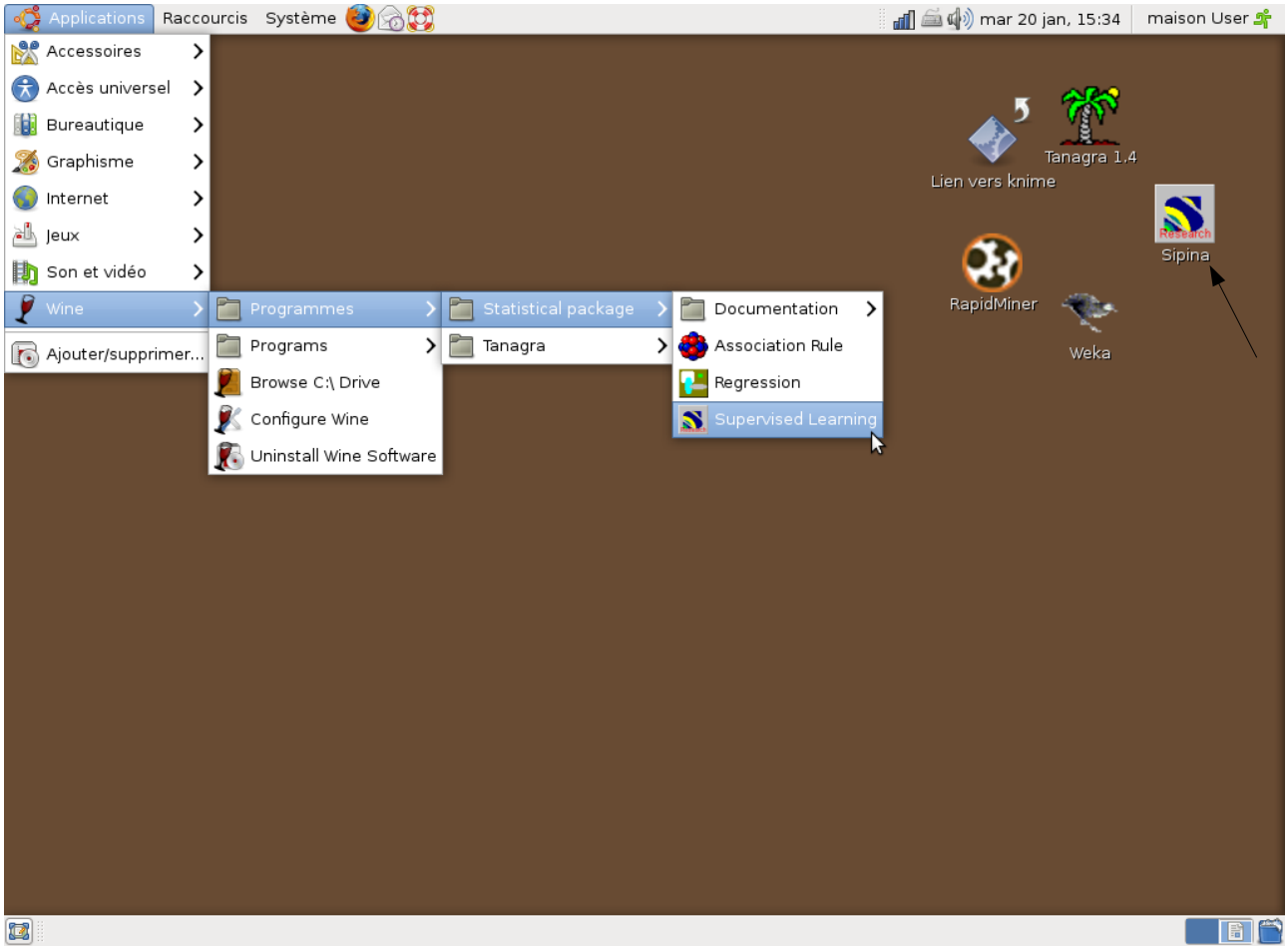
Comme j'ai eu l'occasion de le dire auparavant, plusieurs utilisateurs m'avaient signalés la possibilité de lancer Tanagra sous Linux. Je ne savais pas trop ce qu'il en était pour Sipina. Il s'avère au final que **Sipina est pleinement fonctionnel**.

2.Installation de Sipina

Après avoir récupéré le fichier « setup_statp_ackage.exe » sur le serveur (http://sipina.over-blog.fr/, voir *Télécharger Sipina*), nous effectuons l'installation en l'exécutant via WINE. La procédure est décrite sur le site de Ubuntu (http://doc.ubuntu-fr.org/wine#installation_d_un_logiciel). Le plus simple est d'introduire la ligne de commande suivante dans un terminal : *wine /votre chemin/setup_stat_package.exe*

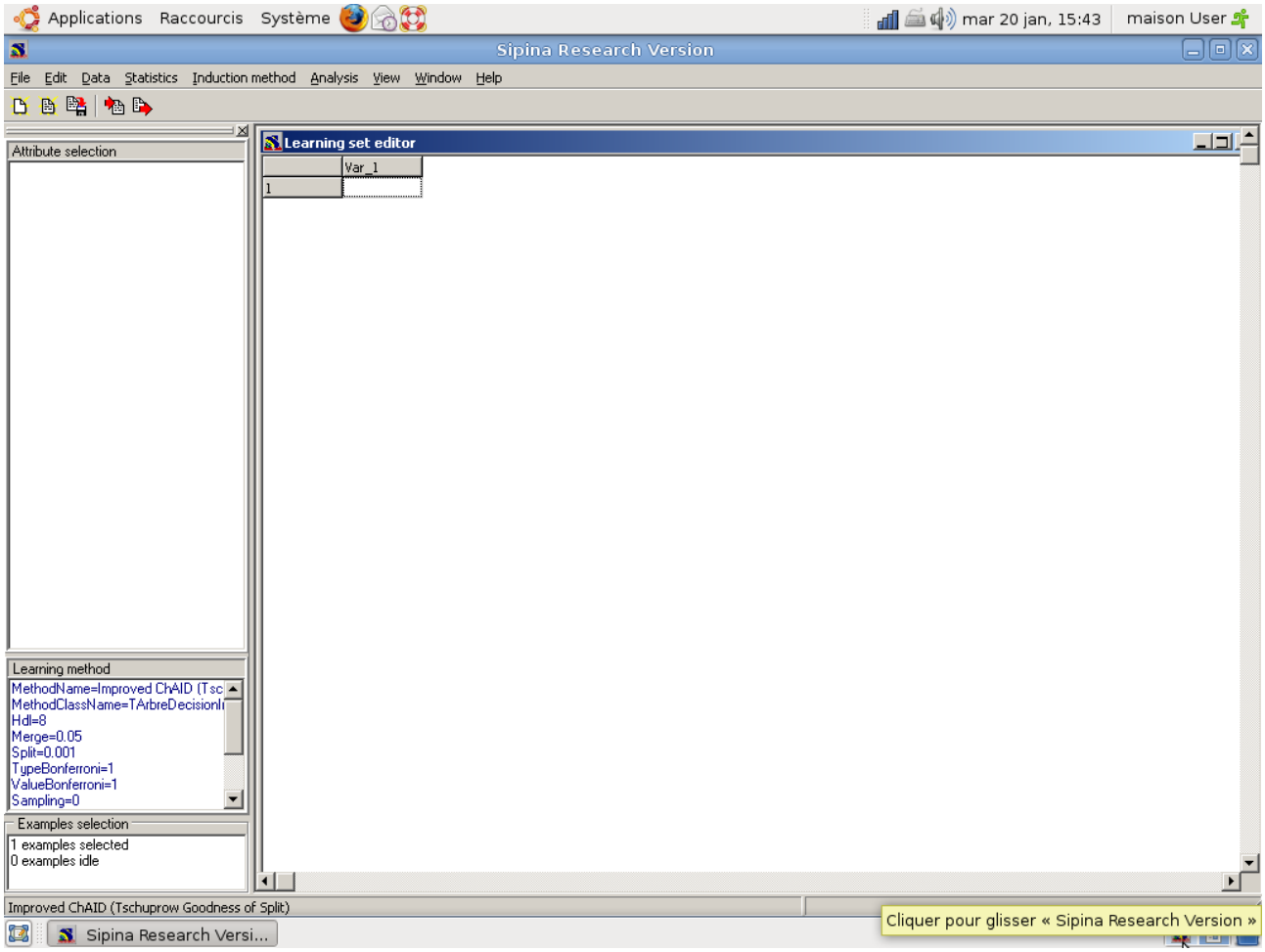
La procédure est démarrée. Il s'agit de valider simplement chaque étape sans trop se poser de questions. Lorsque le programme demande s'il doit installer le logiciel dans le répertoire « c:\program files\... », qui n'existe pas physiquement sur votre disque dur, il ne faut pas s'en formaliser. Le système se charge de copier les fichiers à l'emplacement adéquat.

A l'issue de l'installation, le groupe « Statistical Package » est maintenant disponible dans le menu « Applications » comme nous pouvons le constater dans la copie d'écran ci-dessous (peut être que serez emmené à re-démarrer votre système pour que le groupe soit visible). Sipina correspond à l'item « Supervised Learning ». Nous avons aussi la possibilité de créer un raccourci sur le bureau.



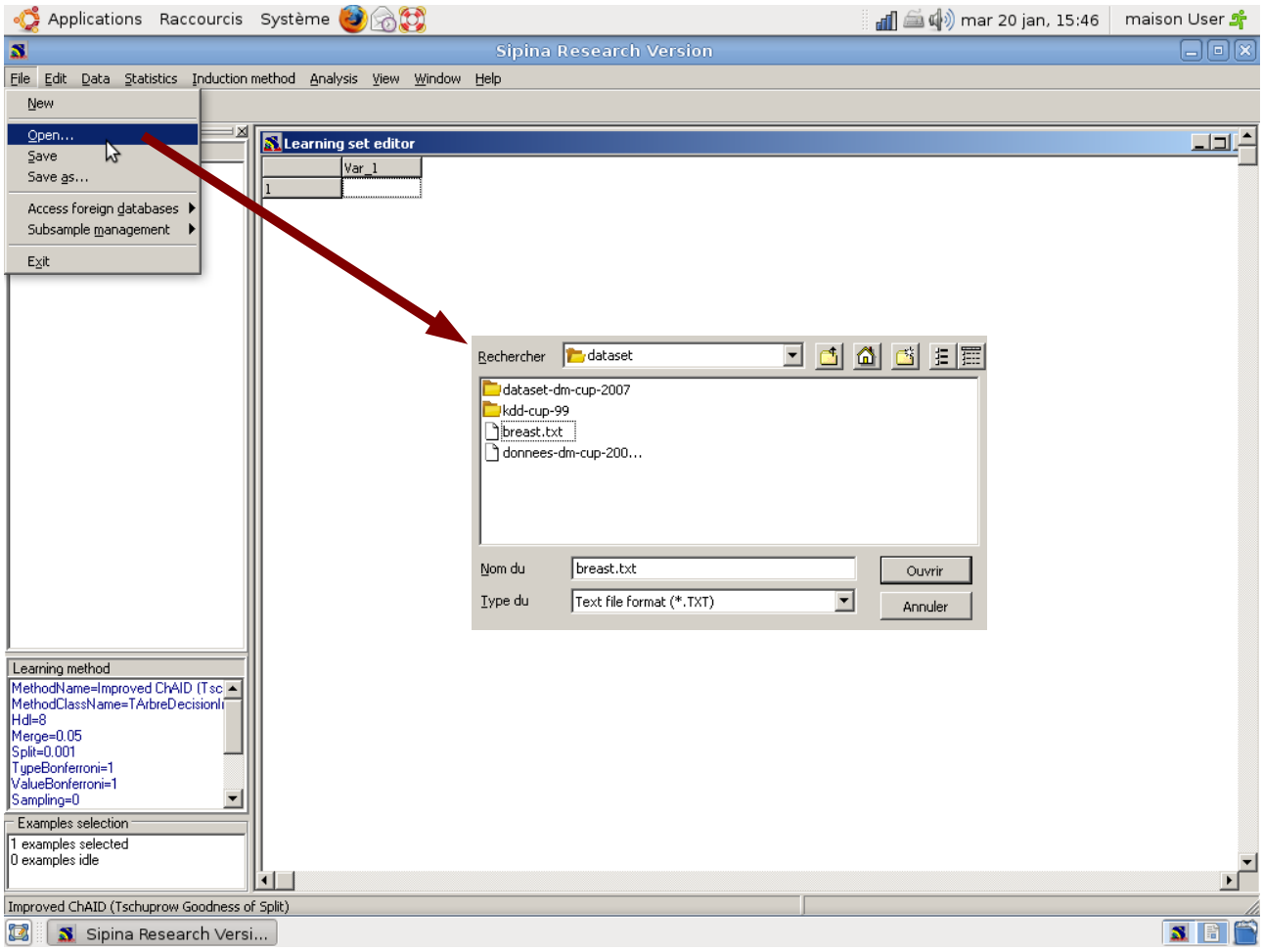
3. Utiliser Sipina

Au démarrage de Sipina, nous retrouvons l'interface habituelle. A vrai dire, rien ne le distingue de son exécution sous Windows.

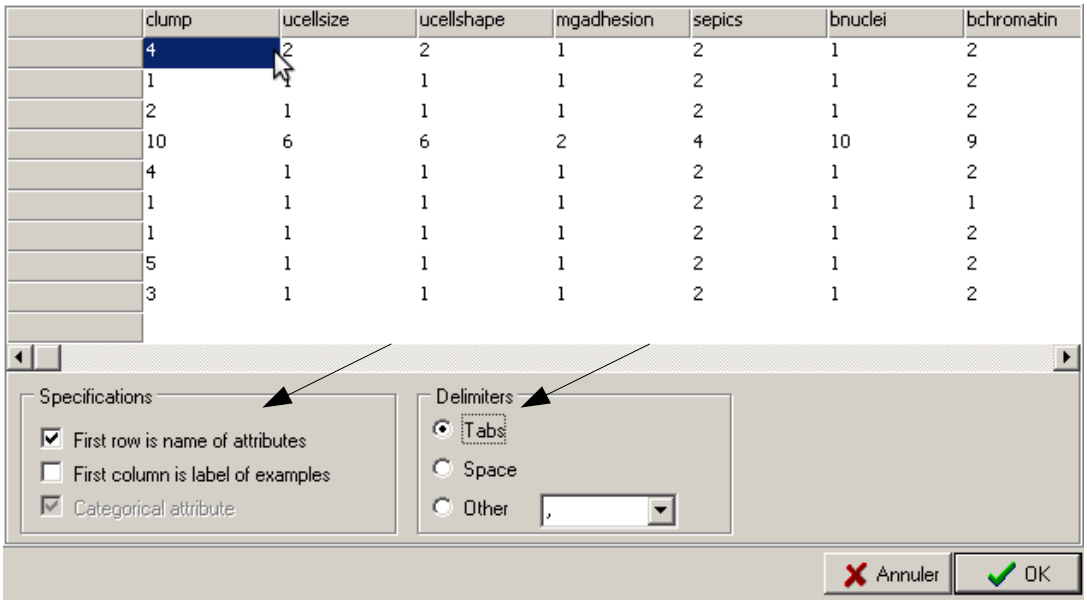


Importation des données. Nous utilisons le fichier BREAST.TXT dans ce didacticiel (http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/breast.txt ; *attention, pour Sipina, le point décimal sera toujours « . » quelle que soit la version de votre système*). Nous souhaitons déterminer le caractère bénin ou malin de cellules extraites d'une tumeur à partir de leur description (taille, forme, etc.).

Pour importer le fichier, nous actionnons le menu FILE / OPEN. Nous optons pour le format « texte avec séparateur tabulation ». Nous sélectionnons « breast.txt ».

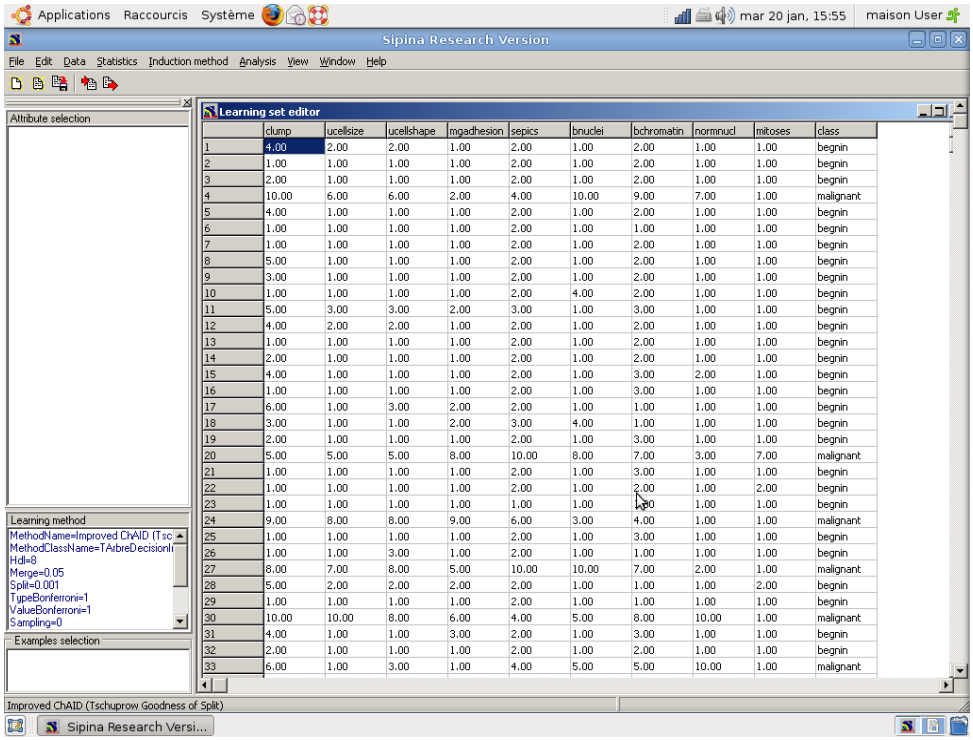


Dans la boîte de paramétrage qui apparaît, nous indiquons que : le séparateur de colonnes est le caractère « tabulation », la première ligne correspond aux noms de variables.

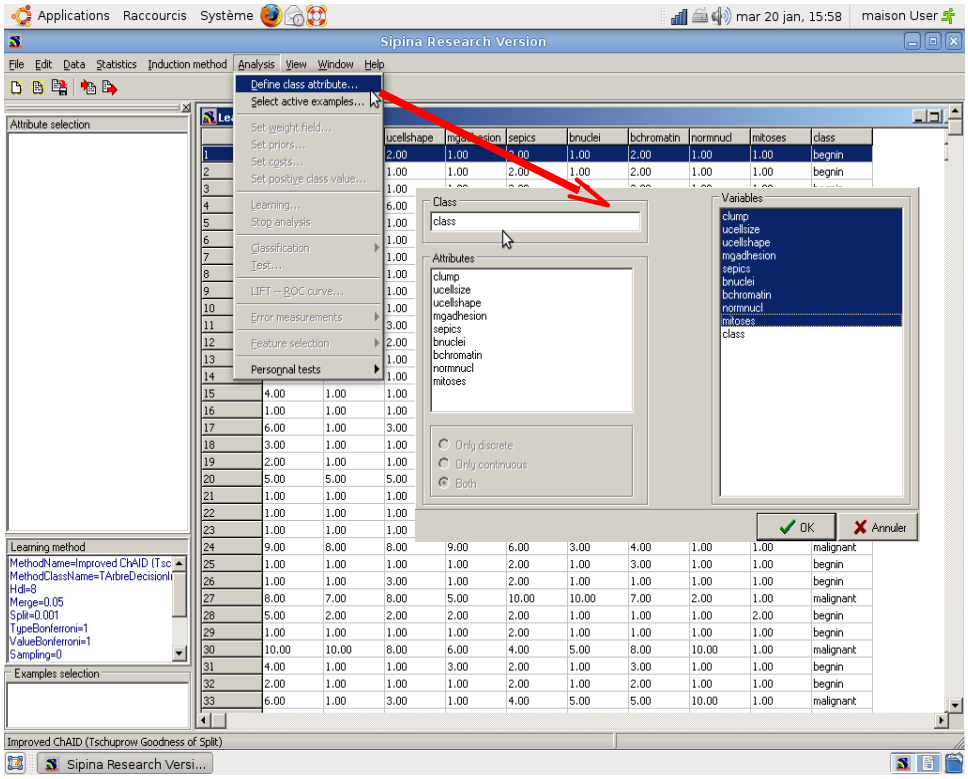


Nous pouvons valider l'opération en cliquant sur OK. Le fichier est importé, il est disponible

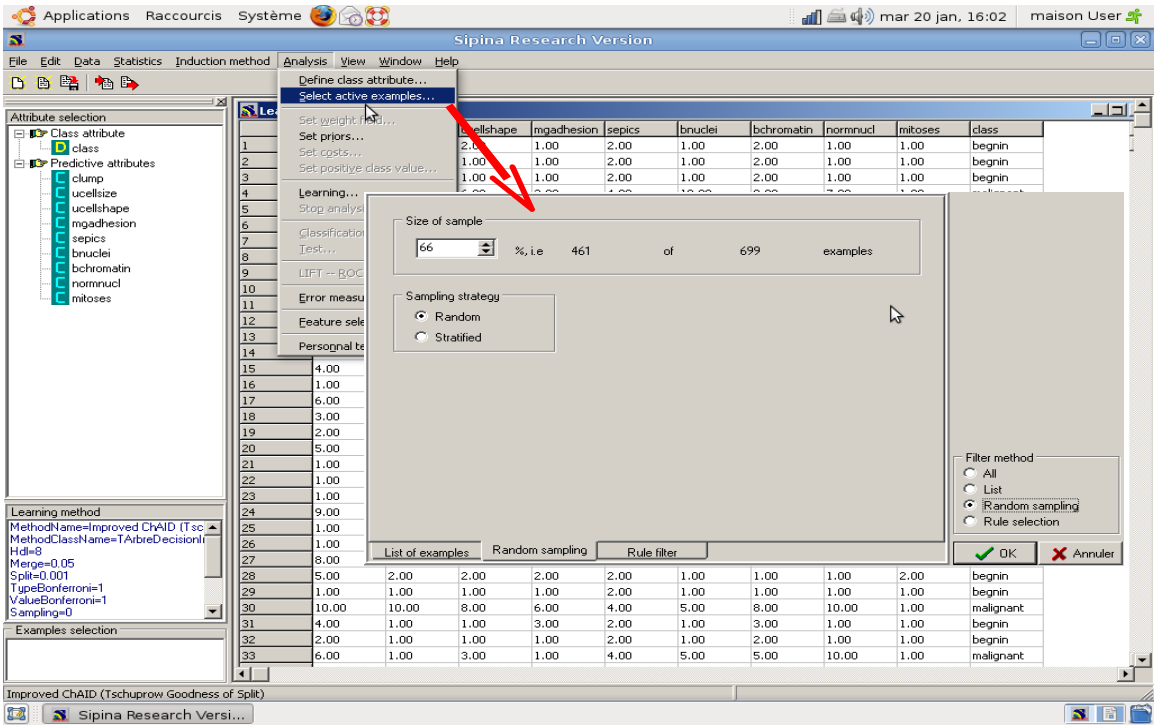
dans la grille de données. Nous pouvons l'éditer le cas échéant.



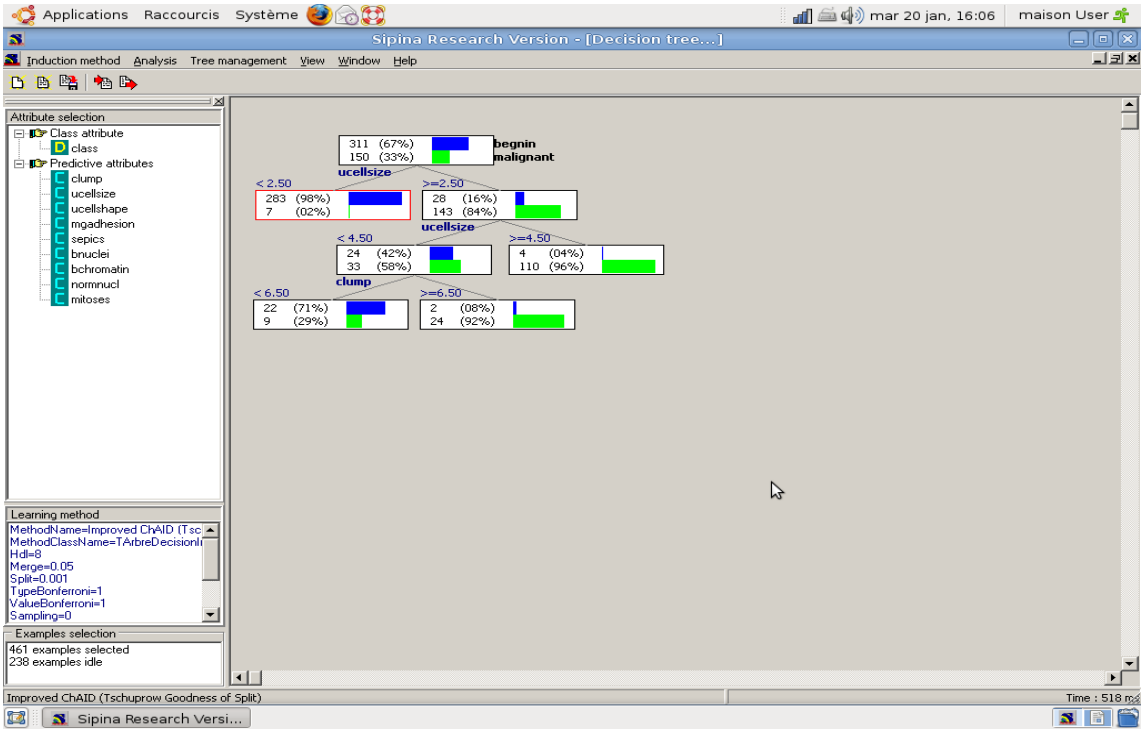
Définir la variable à prédire et les variables prédictives. Nous devons indiquer à Sipina que « class » est la variable à prédire, les autres sont les prédictives. Nous actionnons le menu ANALYSIS / DEFINE CLASS ATTRIBUTE. Par glisser-déposer, nous réalisons le bon paramétrage.



Subdiviser les données en « apprentissage » et « test ». Nous souhaitons réserver 66% des données pour l'élaboration de l'arbre de décision, 34% pour son évaluation. Nous actionnons le menu ANALYSIS / SELECT ACTIVE EXAMPLES. Dans la boîte de paramétrage, nous choisissons l'option RANDOM SAMPLING : 66% sont sélectionnés, soit 461 individus.

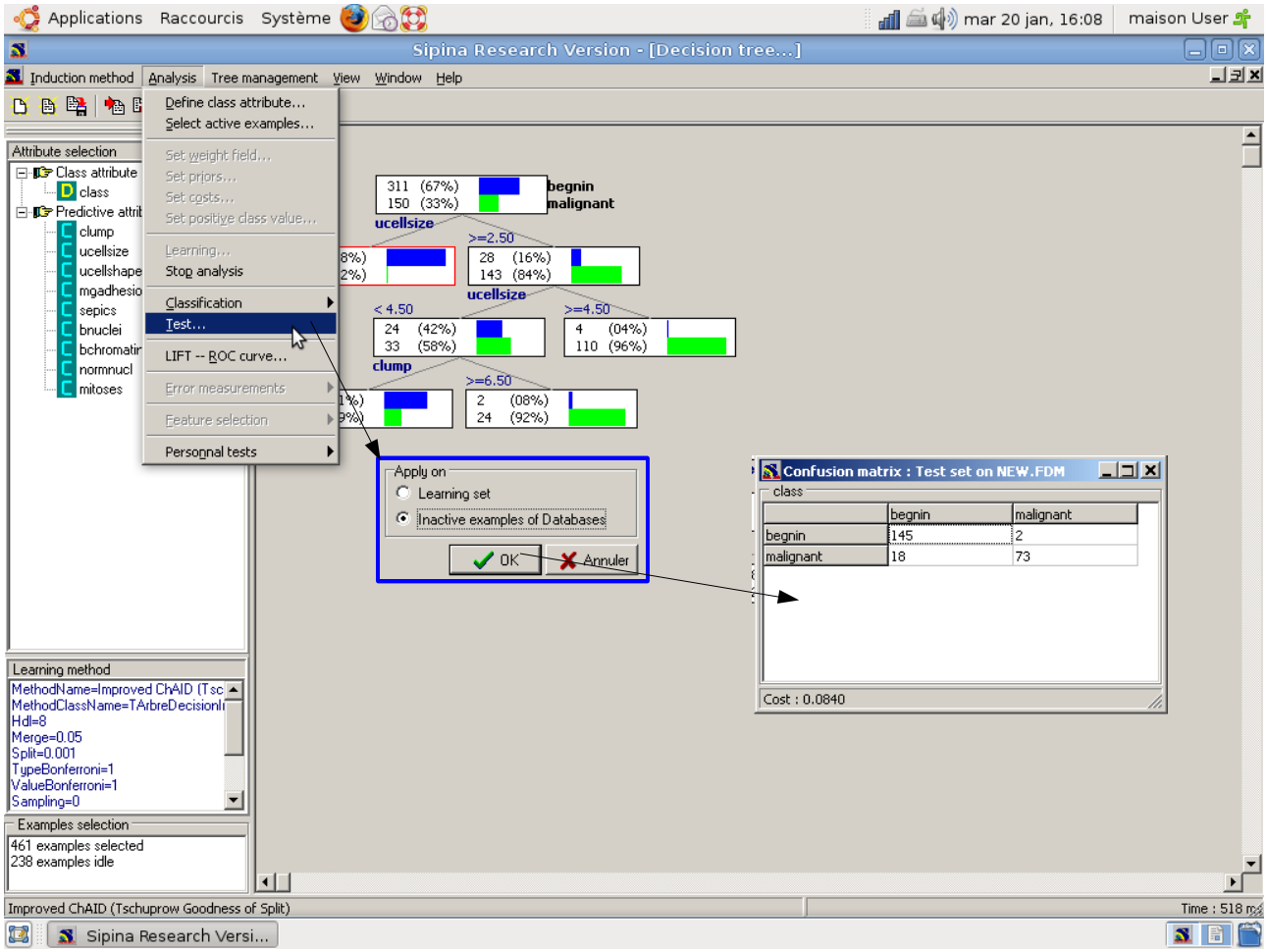


Création de l'arbre de décision. Pour lancer l'apprentissage, nous actionnons le menu ANALYSIS / LEARNING. L'arbre de décision apparaît alors dans une fenêtre dédiée.



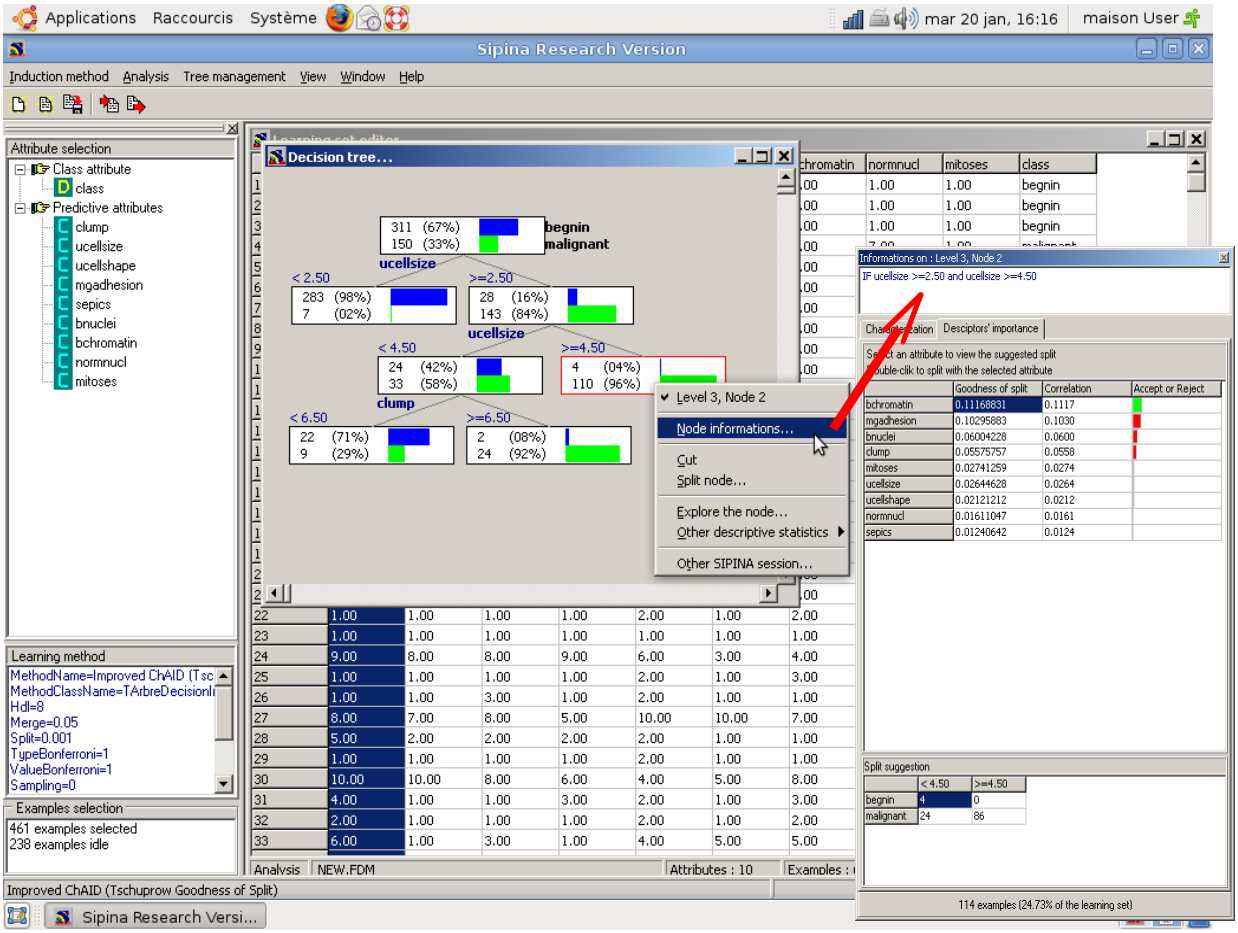
Remarque *: Attention, la subdivision étant aléatoire, il est naturel que nous n'ayons pas les mêmes résultats d'une machine à l'autre, d'une exécution à l'autre*.

Évaluation des performances. Nous devons maintenant évaluer les performances de l'arbre sur l'échantillon test c.-à-d. les 238 observations que nous avons mis de côté. Nous actionnons le menu ANALYSIS / TEST. Dans la boîte de dialogue, nous indiquons une évaluation sur la partie non sélectionnée des données. Une nouvelle fenêtre contenant la matrice de confusion apparaît alors. Dans notre cas, le taux d'erreur en test est de 8.40%.



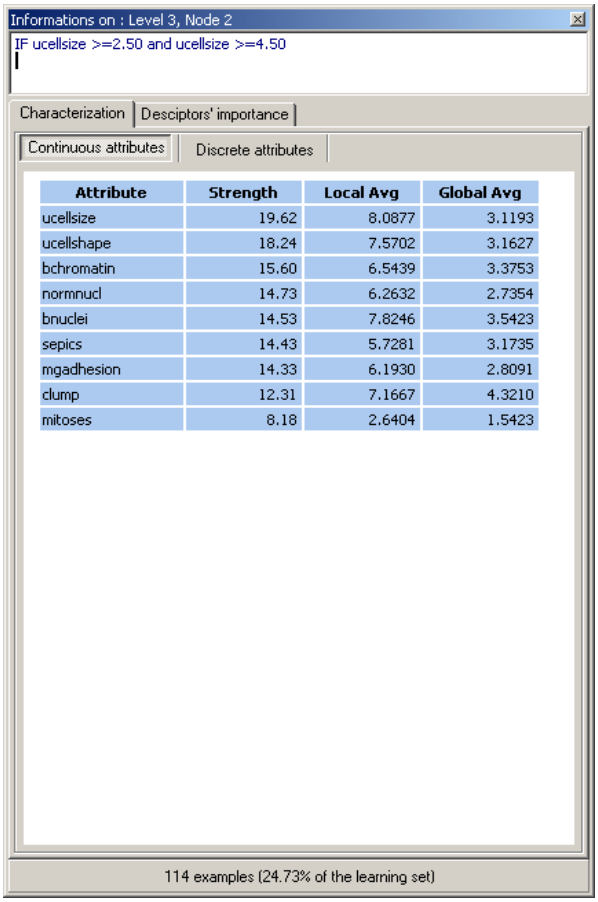
Exploration des sommets de l'arbre. Intéressons nous maintenant à un des sommets de l'arbre de décision, celui à droite avec une majorité de « malignant ». Nous souhaitons l'explorer finement.

Pour ce faire, nous le sélectionnons, via un clic avec le bouton droit de la souris, nous faisons apparaître le menu contextuel. Nous actionnons l'item NODE INFORMATIONS. Une fenêtre d'information apparaît. Dans un des onglets (DESCRIPTOR'S IMPORTANCE), nous avons la liste des segmentations candidates.



Dans l'autre onglet (CHARACTERIZATION), nous observons les statistiques descriptives comparatives qui permettent de caractériser le sous-ensemble d'observations à l'aide des variables qui n'apparaissent pas dans l'arbre.

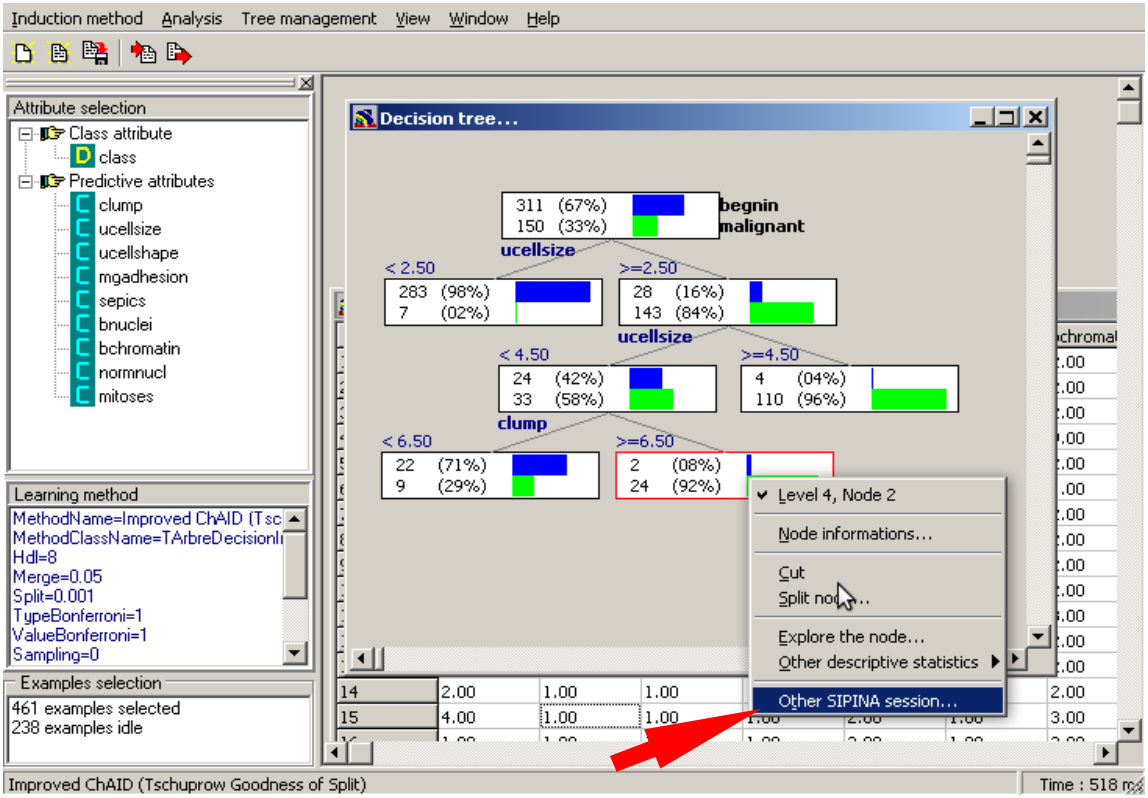
Dans ce exemple, nous constatons entre autres que UCELLSHAPE, qui n'apparaît pas dans l'arbre, tout du moins dans le chemin de la racine au sommet étudié, est aussi une variable caractérisante forte de la sous-population associée au nœud : sa valeur moyenne est nettement plus élevée dans ce sous-groupe (7.57) que dans la globalité du fichier de données (3.16).



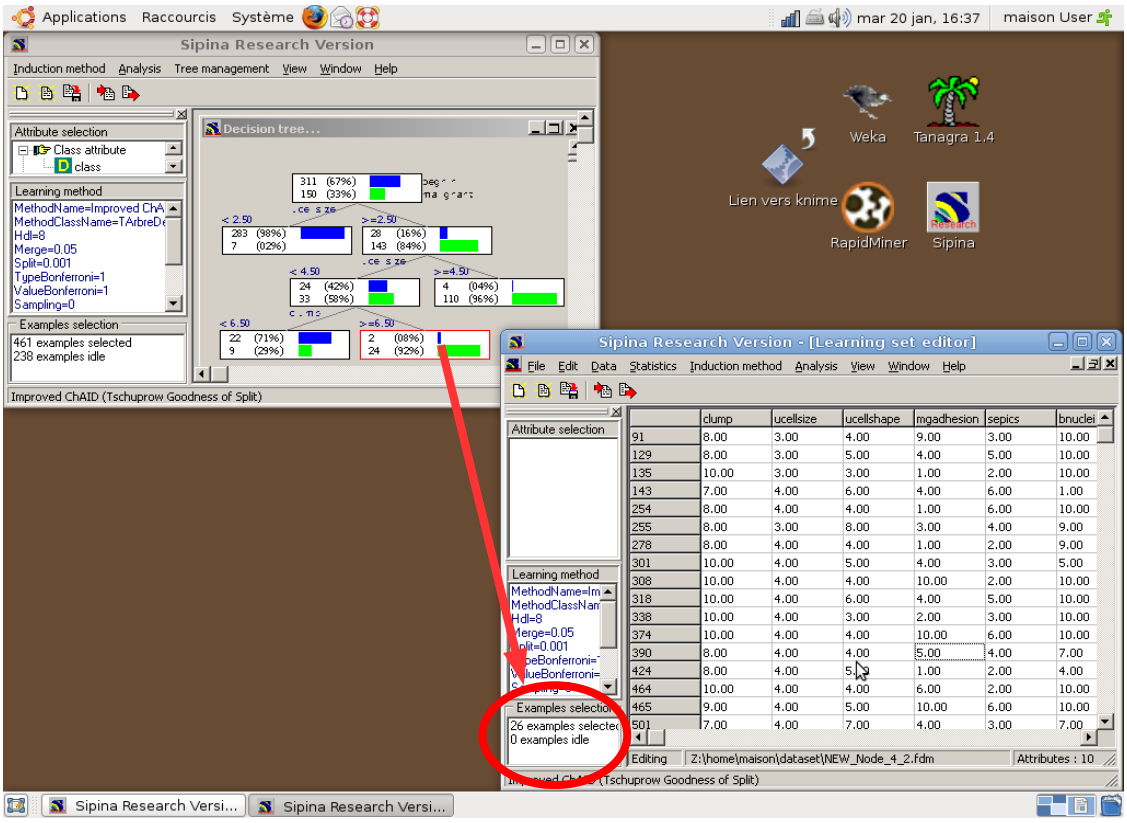
4. Explorer une sous-population

Mettons que nous souhaitons explorer plus finement encore la sous-population décrite par le nœud comportant 26 observations (avec la description « UCELLSIZE >= 2.5 **ET** UCELLSIZE < 4.5 **ET** CLUMP >= 6.5 »)

Toujours via le menu contextuel, nous avons accès à l'item OTHER SIPINA SESSION. Il permet de démarrer automatiquement une autre instance de SIPINA en lui transmettant le sous-fichier associé au nœud.



Dans la nouvelle exécution de SIPINA, nous constatons que seules les observations concernées ont été transmises. Toutes les variables sont en revanche présentes.



Nous pouvons dès lors lancer une nouvelle analyse en spécifiant d'autres paramètres ou en choisissant d'autres variables d'intérêt. Nous pouvons démultiplier à l'infini l'exploration des données.

Remarque : A vrai dire, cette opération ne présente pas de difficultés particulières en programmation. Il est fréquent que l'on démarre automatiquement un logiciel à partir d'un autre. La vraie surprise ici est que l'appel à une API Windows faisant référence au SHELL ait été correctement interprété par Linux, qui s'est chargé de lancer le logiciel via WINE. Je trouve que c'est très fort. Toutes les fonctions d'un logiciel compilé pour Windows sont opérationnelles, y compris celles faisant appel à des procédures de bas niveau.

5. Conclusion

Dans ce didacticiel, nous avons montré qu'il est possible d'installer et d'utiliser Sipina dans l'environnement Linux sans qu'il soit nécessaire d'avoir des compétences systèmes particulières. Nous bénéficions alors de tout l'environnement fonctionnel du logiciel.