

Projet Mistral-RAD

Didier HERMANN / STAGIAIRE
Développeur Logiciel

Stage chez la Société R.Tech / VERNIOLLE

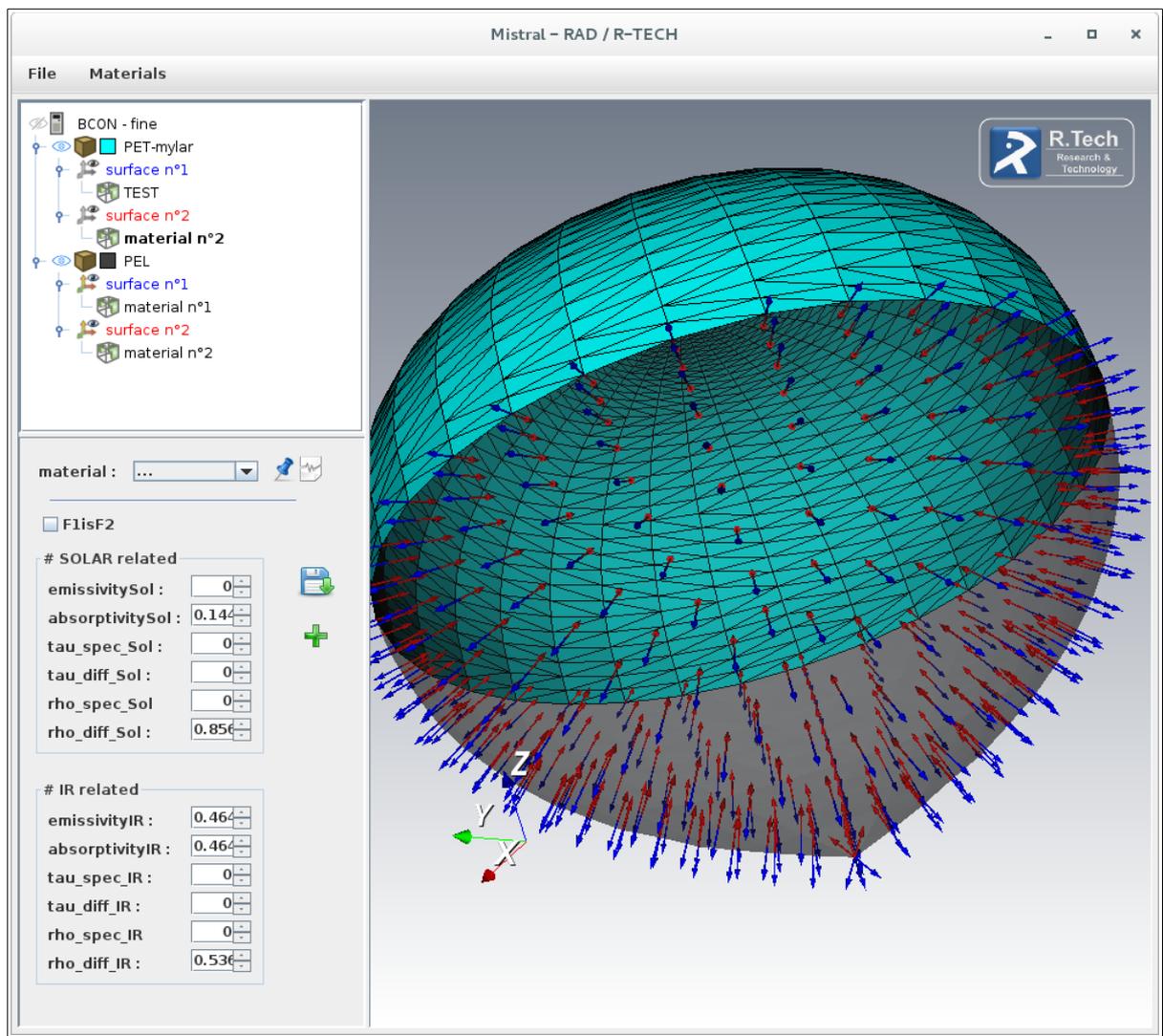


Table des matières

I.Remerciements.....	3
II.Présentation de la société R.Tech.....	4
III.Sujet de STAGE.....	7
IV.Liste des compétences professionnelles validées pendant le stage.....	8
V.Résumé / Abstract.....	8
VI.Expression des besoins.....	9
VII.Spécifications Fonctionnelles.....	14
VIII.Spécifications Techniques.....	15
IX.Présentation d'autres applications.....	16
1.SEIS AUTAN Papagayo.....	16
2.ParaVIEW et Cassandra.....	17
X.Réalisation.....	19
1.Explication et installation de la bibliothèque VTK.....	19
Qu'est ce VTK?.....	19
Avant d'aller plus loin : Comment installer et utiliser VTK avec JAVA dans Eclipse ?.....	19
2.Réalisation de la JavaDoc.....	24
3.Description des Packages et Classes.....	24
4.Exemples de Code et appels de fonctions.....	25
5.Gestion des Matériaux dans le panneau de saisie.....	27
6.Déploiement de l'application.....	30
XI.Améliorations.....	32
XII.Conclusion.....	33
XIII.Annexe Diagrammes de classe des différents éléments du programme.....	34
XIV.Annexe Planning des tâches du projets.....	38
XV.Annexe Guide de l'utilisateur.....	41

Liste des illustrations

organigramme de l'Entreprise (document R.Tech).....	5
première maquette de l'application issue de la réunion du 22/05/2016 (numérisation d'un document de Travail).....	9
copies d'écran de la première réalisation.....	10
schéma de l'interface graphique de l'application (vers.2) (photo du whiteboard) issu de la réunion du 03/05/2016.....	11
copies d'écran et évolution visuelle de l'interface graphique de l'application.....	12
cas d'utilisation de Mistral-RAD.....	14
copie d'écran de SEIS.....	16
température volumique et surfacique d'un Ballon MIR.....	17
les multi-réflexions énergétiques sur un objet.....	17
copie d'écran de ParaView.....	18
copie d'écran Cassandra.....	18
copie d'écran de « ccmake ».....	20
packages et Classes de Mistral-RAD.....	24
gestion de l'association des Matériaux de la base de données XML dans les panneaux de saisie.....	27
étapes d'installation de Mistral-RAD.....	31
image des mots clés.....	33
diagramme / données de conditions limites.....	34
diagramme / interface graphique.....	35
diagramme / éléments VTK.....	36
diagramme / base de données des Matériaux.....	37
planning des tâches réalisées.....	38
détail des tâches réalisées.....	39

I. Remerciements

Tout d'abord, j'adresse un remerciement à M. SPEL le directeur de la société R.Tech et tuteur de mon stage pour m'avoir intégré rapidement au sein de l'entreprise et m'avoir accordé toute sa confiance.

Je remercie toute l'équipe de la société R.Tech pour le professionnalisme et le temps qui m'ont accordé sachant répondre à toutes mes interrogations et qui ont contribué au succès de mon stage et à la réalisation de mon Projet.

Je souhaite ajouter que l'environnement de travail chez R.Tech est agréable, bénéfique et qui permet de travailler dans la plus grande sérénité.

J'adresse aussi un remerciement à LDNR, aux intervenants et leurs co-fondateurs pour leur soutien et le contenu de leur Formation.

II. Présentation de la société R.Tech



R.Tech is an SME specializing in Computational Fluid Dynamics, with expertise in modelling high-speed flows. State-of-the-art computational simulations are performed by using innovative hardware techniques combined with close links to research institutes. R.Tech's main efforts are centred on code development (CFD and engineering tools) as well as large-scale computations, which can be completed in a short turn-around time.

Grid generation for large complex three-dimensional structured grids is an area in which R.Tech is specialised using commercial software such as GridPro. Work in unstructured and hybrid grid generation is currently under development.

The computational power required for the simulation of complex studies is achieved using the R.Tech designed supercomputing facilities. Large-scale clusters of single or multi-processor machines are created to obtain combined CPU power, memory and storage space.

R.Tech also offers services to port and benchmark existing codes on cluster architectures.

After the development of its activities in new domains such as engineering modelisations and creation of intuitive human machine interfaces, the company is always ready to face new challenges and will be the perfect partner for your future innovative projects

R.Tech est une PME créée en 2001 et spécialisée dans la simulation numérique et le développement de logiciel.

Depuis sa création R.Tech travaille dans le domaine de CFD (Computational Fluid Dynamics), avec une expertise dans la modélisation des écoulements à haute vitesse. Les simulations numériques sont effectuées en utilisant des techniques matérielles innovantes associées à des liens étroits avec les instituts de recherche. Les principaux efforts de R.Tech sont centrés sur le développement de code (CFD et outils d'ingénierie) ainsi que des calculs à grande échelle, à exécution rapide grâce à ces importants moyens de calcul.

R.Tech développe des logiciels sur mesure dans le domaine de simulation numérique, vision industrielle, bases de données, suivi de production, visualisation 3D et des segments sols de suivi de ballons.

Moyens en personnel

L'équipe R.Tech est composée de quatre ingénieurs en simulation numérique des fluides ainsi que quatre experts en sciences de l'informatique et une secrétaire. Cet effectif permet d'assurer la continuité des projets vis à vis d'autres entreprises aux effectifs moins réduits.

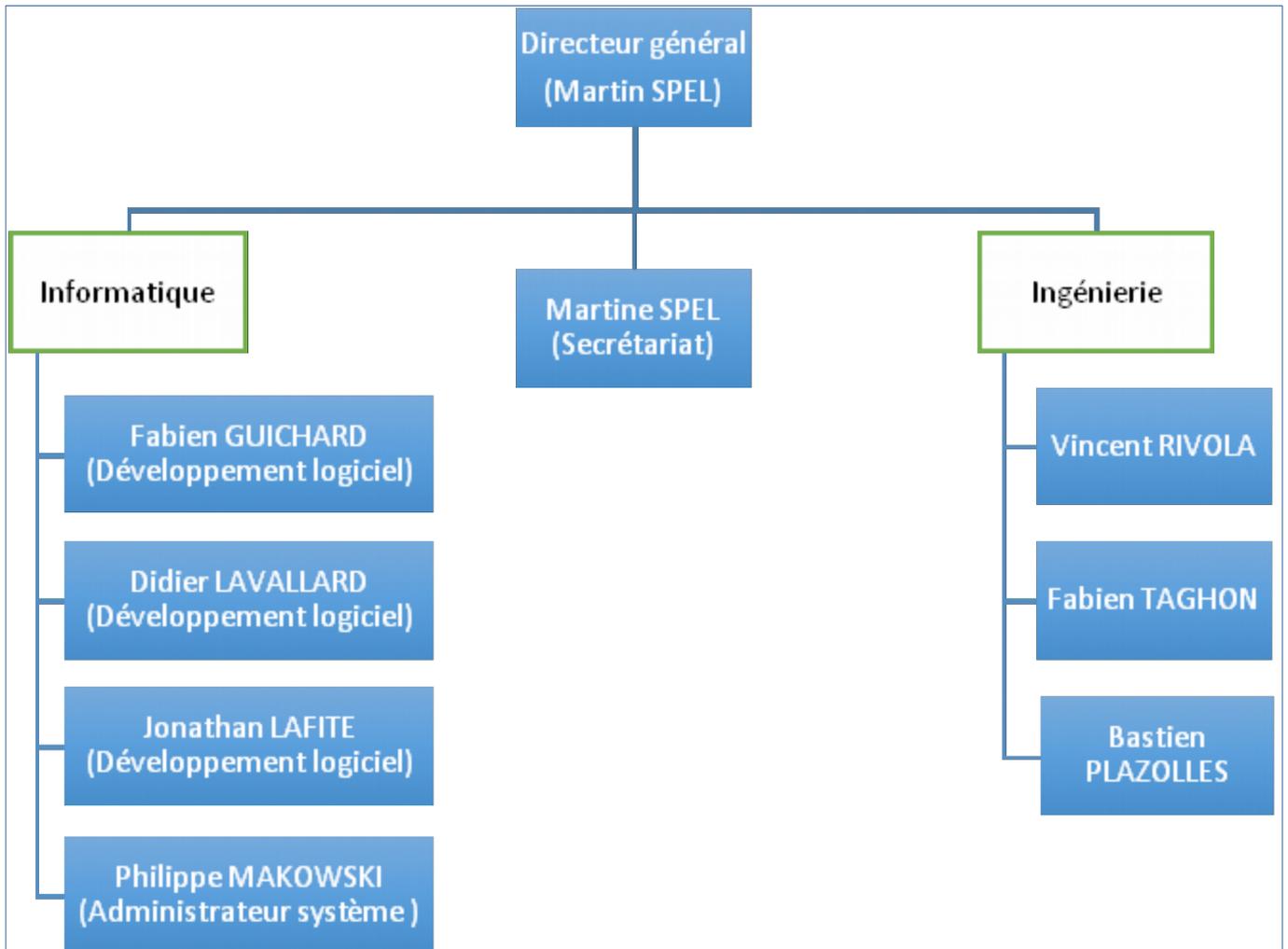


fig. 1: organigramme de l'Entreprise
(document R.Tech)

Les installations de R.Tech sont réparties en trois domaines principaux : infrastructure de **calcul numérique**, **ingénierie** et **développement logiciel**.

Calcul numérique

L'infrastructure de calcul numérique est composée d'un cluster de 240 cœurs avec un total de 640 GB de mémoire. Un ordinateur avec une carte graphique est dédié aux calculs de rayonnement. La salle des serveurs permet d'envisager une extension de l'infrastructure si besoin. Des systèmes de sauvegardes automatiques fournissent une protection contre la perte de données.

Ingénierie

Le secteur « ingénierie » de R.Tech a développé une suite de logiciels permettant de calculer avec une grande fidélité tous les points de trajectoire suivis par un véhicule spatial entrant dans l'atmosphère, en ré-entrée destructive ou non. En complément des outils logiciels développés en interne, un jeu de logiciels open-source est disponible.

Pour le régime raréfié ou transitionnel, il peut être utilisé soit MISTRAL-DSMC, récemment développé, soit dsmcFOAM. C'est un avantage majeur de disposer d'au moins deux codes de calcul afin de pouvoir effectuer des vérifications croisées.

En régime hypersonique, le code MISTRAL-CFD, validé par plusieurs projets européens (parmi lesquels le récent IXV qui a été un succès) peut être utilisé.

MISTRAL-CFD est développé depuis 2001 et permet de faire des simulations aérothermodynamiques, incluant les phénomènes de déséquilibre thermo-chimique et d'ionisation faible. Pour les flux non hypersoniques en régime continu, une suite de trois codes est disponible: MISTRAL, SU2 et OpenFOAM.

Les outils de CAO utilisés sont principalement SolidWorks et Rhinoceros. Ces deux outils permettent la construction de formes géométriques, l'import de différents formats dont STEP et IGES, et l'export vers différents formats parmi lesquels des descriptions de géométries discrétisées tels que le format STL.

Pour le maillage tridimensionnel, le logiciel GridPro est utilisé. GridPro est un générateur de maillage 3D commercial qui permet de créer des maillages hexaédriques complets pouvant être, si on le souhaite, dégradés en maillages tétraédriques.

Parmi les logiciels de visualisation utilisés couramment figurent Tecplot, Paraview et GnuPlot et les outils de rayonnement qui ont été développés Autan et Papagayo.

Développement logiciel

Le secteur « développement logiciel » de R.Tech utilise une vaste gamme d'environnements de développement intégrés (Eclipse, Netbeans, QtCreator, Visual Studio, Windev, Webdev ...) selon les besoins.

En plus du module JUnit pour faire des tests unitaires, Robot Framework (qui permet des tests automatiques d'interfaces graphiques) est utilisé pour s'assurer que les composants graphiques sont eux aussi exempts de bogues.

Des outils développés en interne pour automatiser la qualité et la traçabilité du code source ainsi que la gestion de configuration des documents sont un précieux atout pour répondre aux standards de qualité.

R.Tech effectue soit du développement logiciel sur demande des clients (simulations d'aérostats, segments sol, outils de simulations thermiques, vision industrielle et outils de prédiction de survivabilité en ré-entrée atmosphérique - Debrisk, Pampero), soit des développements internes comme MISTRAL-CFD ou des outils de simulations thermiques, de gestion documentaire, ...

III. Sujet de STAGE

Au sein de l'entreprise R.Tech un logiciel a été développé pour évaluer les apports énergétiques solaires et infrarouges terrestres sur un aérostat stratosphérique.

L'outil prend en compte la transmissivité, les multi-réflexions et le rayonnement surface-surface par le biais d'une méthode de lancer de rayons.

Actuellement l'outil prend en entrée un jeu de fichiers de surfaces décrit par un ensemble de triangles connectés, un fichier de paramètres numériques, et un fichier de conditions limites (propriétés thermo-optiques, flux solaires et infrarouge).

Chaque fichier correspond à un sous-ensemble de la géométrie ou « patch » où il faut imposer pour chaque côté les propriétés thermo-optiques.

Actuellement ces fichiers sont écrits à la main avec un éditeur de texte, ce qui ne permet pas d'identifier visuellement les différents faces des objets, et qui peut induire des erreurs.

Dans le cadre du stage un outil a été développé qui permet à l'utilisateur de :

- x importer les fichiers au format VTK, STL et OBJ
- x afficher les différents patches en 3D
- x assigner des différentes conditions limites à chaque coté des patches
- x écrire un fichier d'entrée avec les conditions limites et les paramètres numériques

L'outil est développé en Java en s'appuyant sur la librairie VTK* pour l'affichage 3D.

L'objectif du projet est de réaliser une interface graphique et de gérer aussi une base de données de matériaux et leurs propriétés de réflexion pour éviter la redondance de la saisie.

* (voir explication et installation de la bibliothèque VTK)

VTK (pour Visualization ToolKit en anglais) est une Bibliothèque logicielle libre de classes C++ pour la visualisation de données 2D/3D et le traitement d'image.

IV. Liste des compétences professionnelles validées pendant le stage

- 1) Développer une application client-serveur
 - Maquetter une application
 - ~~Concevoir une base de données~~
 - ~~Mettre en place une base de données~~
 - Développer une interface utilisateur
 - Développer des composants d'accès aux données
- 2) Développer une application web
 - ~~Développer des pages web en lien avec une base de données~~
 - ~~Mettre en œuvre une solution de gestion de contenu ou e-commerce~~
 - ~~Développer une application simple de mobilité numérique~~
 - Utiliser l'anglais dans son activité professionnelle en informatique
- 3) Compétences transversales de l'emploi
 - Actualiser et partager ses compétences en développement informatique
 - Organiser son activité en développement informatique

V. Résumé / Abstract

Within the company R.Tech software was developed to assess the terrestrial solar and infrared energy intake on a stratospheric balloon.

The tool takes into account the transmissivity, multi-reflections and radiation surface to surface by means of a method of ray tracing.

Currently the tool takes as input a set of files surfaces described by a set of connected triangles, file numerical parameters, and file boundary conditions (thermo-optical properties).

Each file corresponds to a subset of the geometry or "patch" which must be imposed for each side thermo-optical properties.

Currently these files are manually written with a text editor, which does not allow to visually identify the different faces of objects, and might cause errors.

As part of a training tool has been developed that allows the user to:

- x import files to VTK format STL and OBJ
- x display different 3D patches
- x assign different boundary conditions at each side of the patches
- x write an input file with the boundary conditions and numerical parameters

The tool is developed in Java based on the VTK library for the 3D display.

The project objective is achieving a graphical interface and also manage a list of materials and their reflective properties to avoid redundant entry .

VI. Expression des besoins

Le suivi du projet tient compte des modèles agiles et se fait sous forme de réunions et de démonstrations du produit qui permet d'avoir une vision réel du résultat final et d'élaborer une critique constructive pour émettre des améliorations et correctifs.

Une première réunion a permis de formuler les besoins et les attentes.

Il est demandé de concevoir une interface graphique pour visualiser des objets en 3D et de pouvoir leur associer des données (conditions limites ou 'Boundary Conditions').

Cette association d'objet/données doit être organisé sous forme de liste car il peut avoir plusieurs objets.

Les données qui ont été collectées sont exportées sous forme d'une fichier texte pour qu'elles puisse être utilisées par les applications « AUTAN » / « Papagayo » développées par R.Tech.

Ci-dessous les premières Étapes de conception après la réunion de lancement du 22 avril 2016 :

- ✓ Création de la fenêtre principale et création d'un tableau qui contiendra les objets que l'on veut traiter dans le fichier bcon-fine
- ✓ Gestion du nom et emplacements des objets
- ✓ Supprimer un élément du tableau et initialisation du tableau
- ✓ Gestion du tableau et des objets : ajouter, supprimer et modifier les éléments du tableau (événements / boutons / affichage de boites de dialogue)
Possibilité de :
 - ⇒ affecter un nom à l'objet
 - ⇒ choisir le fichier 3D de l'objet
 - ⇒ visualiser l'objet
 - ⇒ choisir la couleur
 - ⇒ aller dans l'interface de paramétrage des conditions limites
 - ⇒ de supprimer l'élément
- ✓ Visualisation fichier .vtk, .stl ou .obj
- ✓ Affichage des objets, gestion des normales et leurs inverses
- ✓ Gestion de l'apparence des objets et leur couleur
- ✓ Vérification des objets ajout d'exception pour récupérer les messages d'erreur
- ✓ Création de la boîte de dialogue pour les propriétés bcon (« Boundary config »)
- ✓ Création d'onglet pour la gestion des faces (face 1 et 2)
- ✓ Mise en place de couleur par défaut
- ✓ Sauvegarde et restauration d'un travail en cours (fichier avec l'extension .bcon)
- ✓ Génération du fichier bcon-fine (possibilité de récupérer un fichier bcon-fine)

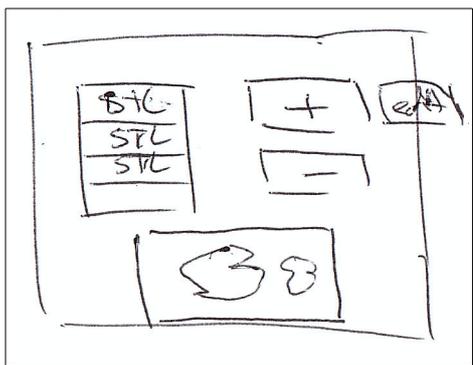
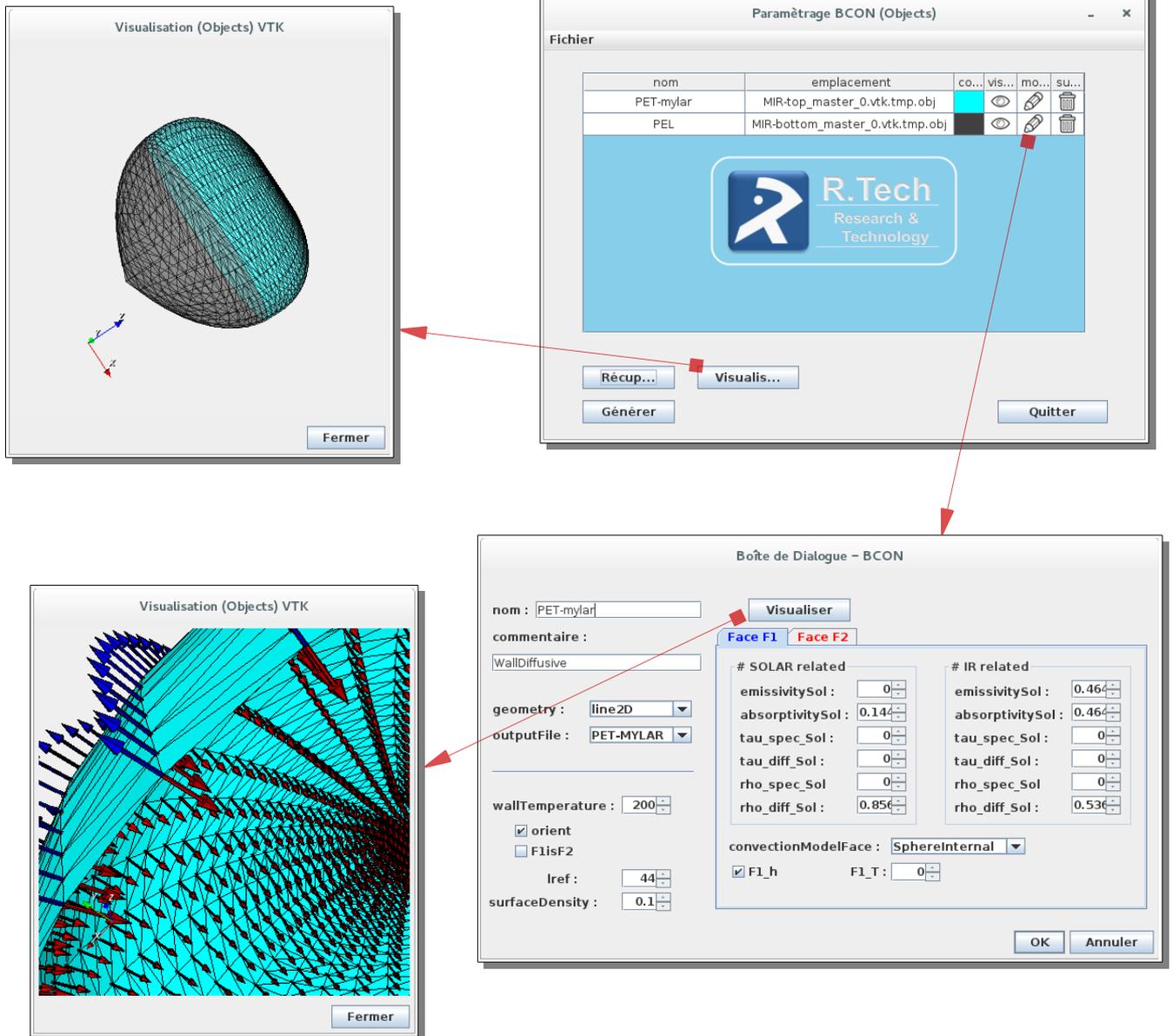


fig. 2: première maquette de l'application issue de la réunion du 22/05/2016 (numérisation d'un document de Travail)

Une première version de l'application a été rapidement réalisée (voir fig.4 copies d'écran) et présentée à l'équipe ingénierie le 3 mai 2016.

A partir de ce moment, nous avons conçu les classes primordiales des différents éléments du programme (éléments VTK et paramètres BCON/conditions limites) nous pouvons commencer à attaquer la réalisation de l'application en prenant en compte les demandes plus concises qui ont suivies la présentation.

fig. 3: copies d'écran de la première réalisation



Le 3 mai 2016, une première version de l'application a été réalisée et présentée. Ci-dessous les nouvelles orientations et recommandation préconisées :

- ✓ Réalisation d'une interface utilisateur avec trois volets (affichage splitté) (voir ci-dessous la photo du whiteboard)
 - ⇒ un volet pour visualiser les objets en 3D
 - ⇒ une arborescence contenant les éléments du fichier bcon-fine :
 - les forme(shapes), matériaux et faces
 - ⇒ un volet de saisie pour chaque sous ensemble
- ✓ Gestion dans l'arborescence du masquage ou affichage d'objets de façon individuelles.
- ✓ Gestion de l'apparence des objets dans le panneau de visualisation 3D.
- ✓ Gestion d'une base de données (au format XML) des matériaux qui permet de pré-remplir les propriétés de réflexion des matériaux.
- ✓ Pouvoir affecter un matériau (un élément de la base de données) à la surface.
- ✓ En haut de la fenêtre mettre en place une barre de menu : File | Shapes | Materials
- ✓ La gestion des matériaux (base de données XML) se fera dans une fenêtre indépendante.
- ✓ Pour la GUI, utilisation les éléments de l'application « SEIS » (voir paragraphe « Présentation d'autres applications »)
- ✓ Autre si le temps le permet vérification des valeurs des matériaux ou valeurs saisies dans le panneau de saisie

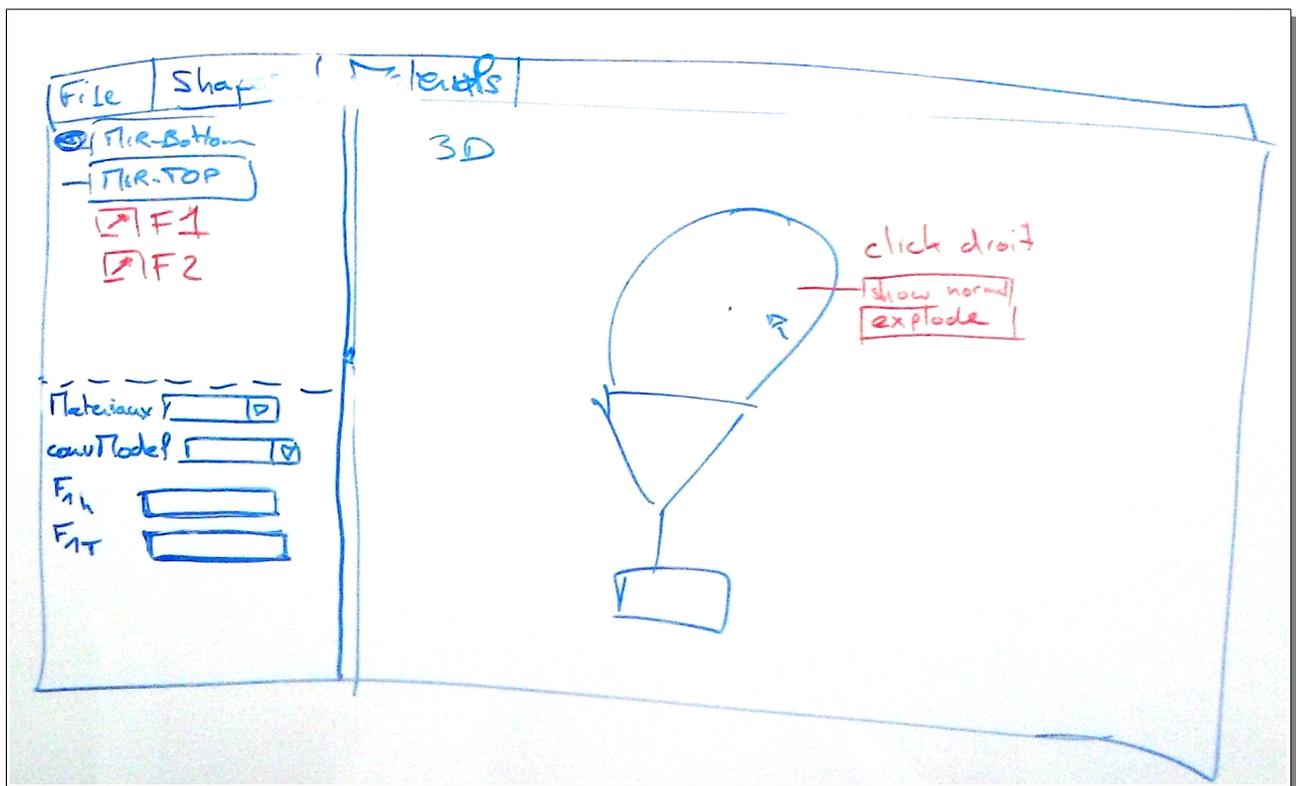
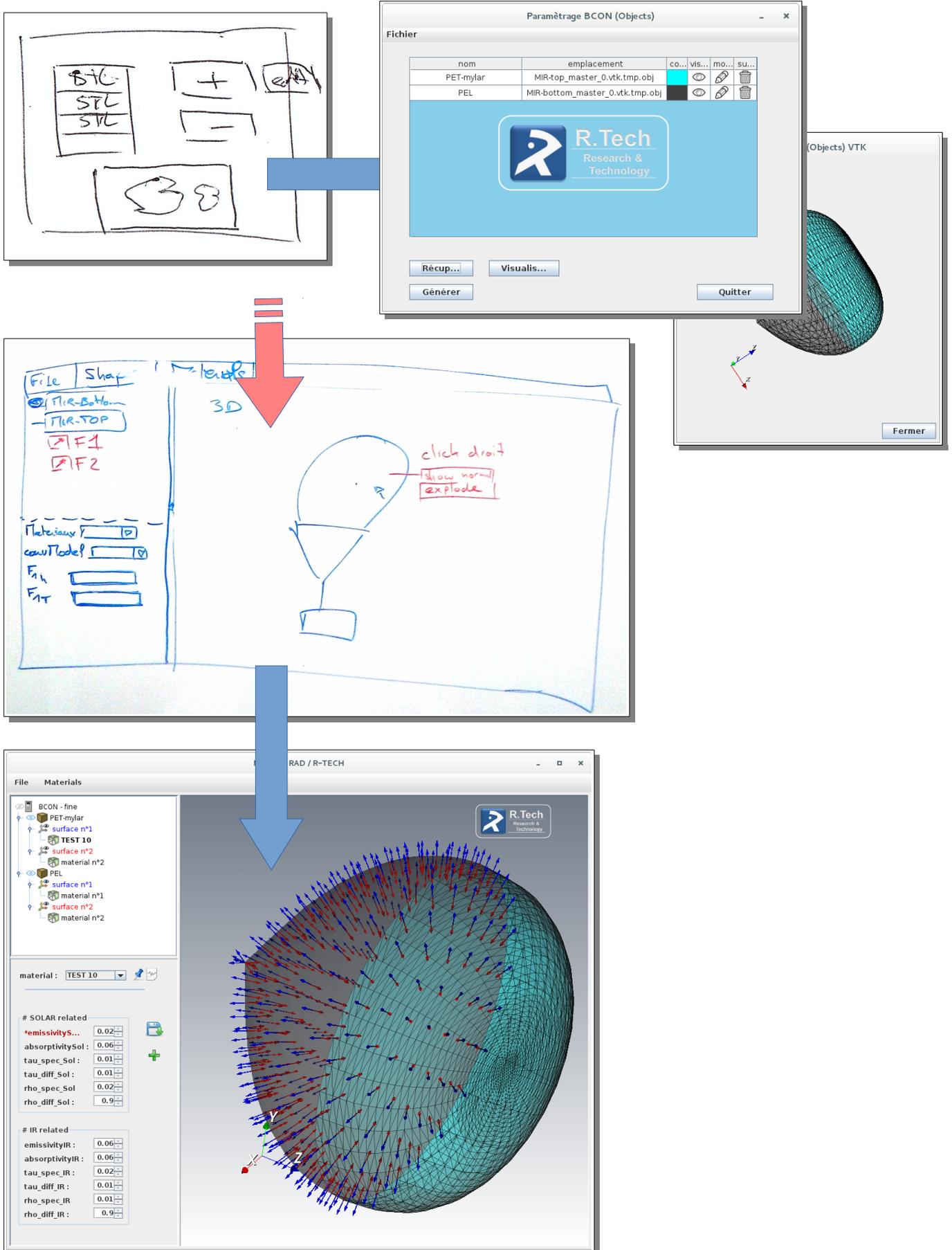


fig. 4: schéma de l'interface graphique de l'application (vers.2) (photo du whiteboard) issu de la réunion du 03/05/2016

fig. 5: copies d'écran et évolution visuelle de l'interface graphique de l'application



Ce qui suit, ce sont les modifications réalisées après les réunions du 18 mai et 30 mai :

(18/05/2016)

- ✓ Vérification des valeurs des matériaux :
 - ⇒ tester si la somme de valeurs ci-dessous est égale à 1 :
 $\text{absorptivity} / \text{tau_spec} / \text{tau_diff} / \text{tau_diff} / \text{rho_spec} / \text{rho_diff}$
 - ⇒ vérifier si **emissivityIR** et **absorptivityIR** sont identiques
 - ⇒ s'assurer si **emissivitySol** soit égal à zéro

Concernant le premier test, on ne peut pas enregistrer les valeurs si la somme est différente de 1
- ✓ Ajouter un n° version dans le fichier XML.
- ✓ Inverser les niveaux (level) dans l'arborescence : Surface/Matériau.
- ✓ Affichage des paramètres matériau dans le volet de saisie.
Quand on associe le matériau à un matériau du fichier XML, indiquer si les valeurs sont différentes.
Ajouter un bouton Save et Add dans le volet de saisie (possibilité de modifier ou ajouter un matériau à partir du panneau de saisie) .
- ✓ Ajouter un code couleur dans l'arborescence pour distinguer les surfaces (rouge ou bleu) .
- ✓ Dans le volet 3D utiliser un fond d'écran avec un couleur dégradée

(30/05/2016)

- ✓ Affecter une hauteur minimum au volet de saisie (pour afficher l'ensemble des valeurs des matériaux à l'écran) .
- ✓ Si les valeurs du matériau sont différentes de celles du matériau avec lequel on l'a associé, on ajoute une étoile devant le libellé du paramètre.
- ✓ Changer la couleur du libellé (en rouge) des valeurs du matériau si celles-ci sont erronées
- ✓ Quand on ajoute un nouveau matériau, demander le nom du matériau que l'on veut ajouter au fichier XML
- ✓ Bug quand on veut sauvegarder un 'Material' avec les JSpinner
(résolu, il faut committer l'éditeur)
- ✓ Le nom du matériau est celui du matériau avec lequel on l'a associé
- ✓ Enlever dans le volet de saisie le bouton 'Refresh'
- ✓ Pour le matériau n°2, ajouter une case à cocher 'F1isF2', si cette case est cochée on rend inactive la saisie des valeurs du matériau. Cela indique que les valeurs du matériau n°1 sont identiques au matériau n°2
- ✓ Dans l'arborescence on peut ajouter un objet uniquement quand on se situe au niveau de la racine de l'arborescence (lors de l'affichage du Popup menu avec clic droit de la souris).
- ✓ Nom de baptême de l'application [MISTRAL-RAD](#)

VII. Spécifications Fonctionnelles

voir en annexe « Guide de l'utilisateur »

Avant d'aller plus loin, il est important de souligner que l'utilité de l'application est de pouvoir identifier les sous-ensembles et surfaces de l'objet que l'on doit traiter.

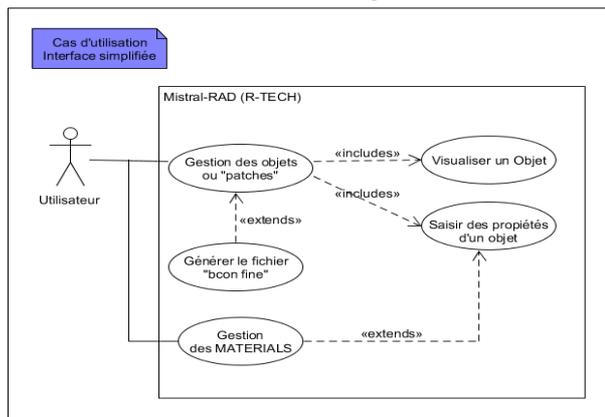
L'objet est défini par un ensemble de « patches », qui eux mêmes sont définis par un ensemble de triangles et chacun d'eux possédant deux FACES.

On doit imaginer une interface qui permet à la fois de visualiser l'objet que l'on traite en 3 dimensions d'où l'utilisation de VTK et de pouvoir affecter à l'ensemble des « patches » des propriétés numériques spécifiques d'où la nécessaire de créer une interface ou panneau de saisie.

Comme certaines données que l'on doit affecter sont assigné à une FACE, il faut identifier précisément les FACES du « patch » d'où l'utilisation de couleur et de normales (pour la partie de visualisation de l'objet 3D).

Puis l'objectif de l'outil est de générer un fichier texte avec une mise en forme spécifique des données. Ce fichier est nécessaire pour le fonctionnement des application AUTAN/Papagayo (applications conçues par R.Tech voir plus loin)

fig. 6: cas d'utilisation de Mistral-RAD



Remarques : ils est nécessaire de gérer plusieurs types de format pour les objets 3D
..... (.stl .vtk .obj) .

Concernant l'interface, on s'est orienté vers une écran splitté avec plusieurs panneaux pour avoir une vision d'ensemble des objets : une arborescence, un panneau de visualisation des objets 3D et un panneau de saisie. Ce qui implique une interrelation concomitante des panneaux. (Je me suis inspiré pour le visuel et la réalisation de l'application SEIS conçue par R.Tech)

Pour alléger la saisie des matériaux, on a ajouté la gestion d'une base de données XML qui permet d'associer les valeurs de ces Matériaux à une FACE.

Comme toute application, on a la possibilité de sauvegarder un travail en cours et d'ouvrir un travail déjà enregistré. On a la possibilité aussi de pouvoir récupérer les données d'un fichier d'entrée Autan déjà généré.

VIII. Spécifications Techniques

L'outil doit être développé en Java en s'appuyant sur la librairie VTK pour l'affichage 3D.

L'outil doit être multiplate-forme .

L'utilisation de Java et VTK sont dans la continuité des précédente réalisation de l'équipe de R.Tech

On utilise la bibliothèque JDOM pour manipuler des données XML.

La résolution minimale est de 998x800.

Version des outils de développement :

- x Java SE Development Kit 8 jdk1.8.0_77
- x Eclipse Mars.2 Release (4.5.2)
(ObjectAid Class Diagram 1.1.10 / Swing Designer 1.8.0.r45)
- x The Visualization Toolkit VTK 7.0.0
- x JDOM 2.0.6
- x Cassandra-2.5.6 + package VTK 5.8.0

Détail concernant l'environnement du poste de travail :

Windows Vista SP2

PentiumD 3.00 GHz + NVIDIA GeForce 7500LE

2Go RAM

Scientific Linux 7.2 + Win 10

Intel Core i3 2.1 GHz et Intel HD Graphics 5500

4 Go RAM

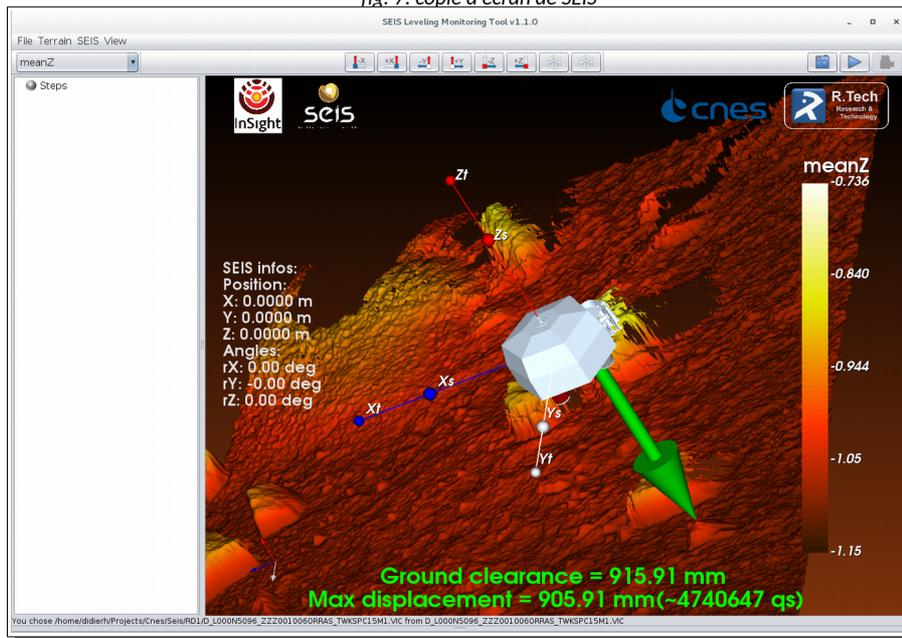
IX. Présentation d'autres applications

Je détaille les logiciels ci-dessous pour que l'on puisse mieux comprendre mon projet. Ces logiciels m'ont servis d'exemple ou m'ont aidé pour réaliser mon Projet.

1. SEIS AUTAN Papagayo

SEIS est application réalisées par R.Tech qui m'a aidé à représenter des objets en 3D et à structurer mon code.

fig. 7: copie d'écran de SEIS



L'outil est réalisé avec la bibliothèque VTK en JAVA et permet de visualiser en 3D un sol Martien et de simuler le levage et dépôt d'un module sur le sol martien par un bras articulé (non représenté dans la simulation).

Les 3 pieds sont munis d'un vérin et on peut jouer sur leur longueur. On déplace au plus proche du sol le module et on fait en sorte que les trois pieds touchent le sol (à partir de cette étape, le bras articulé n'intervient plus).

Le but de l'outil est de rapprocher le dessous du module le plus près du sol en agissant sur la longueur des pieds et en veillant toujours à sa stabilité.

L'application enregistre la position du module ainsi que la longueur des pieds que l'on affecte.

L'outil permet donc de visualiser et surtout de tester étape par étape le dépotage du module sur le sol martien.

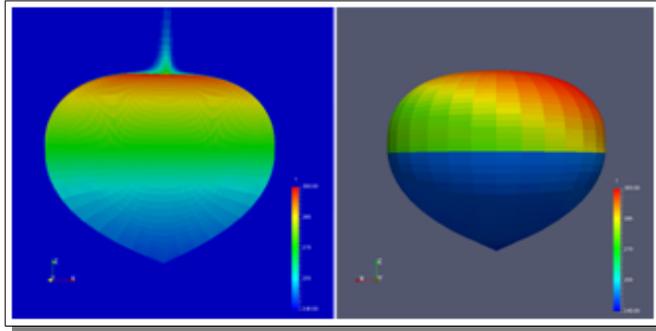
Les deux autres applications sont à l'origine de mon projet : c'est l'outil à développer qui fournit les éléments d'entrées nécessaires (« bcon fine »)

pour **AUTAN** qui permet de modéliser les phénomènes de radiation sur un objet et

pour **PAPAGAYO** qui permet de calculer de manière itérative les températures des différents « patches » issu de **AUTAN**.

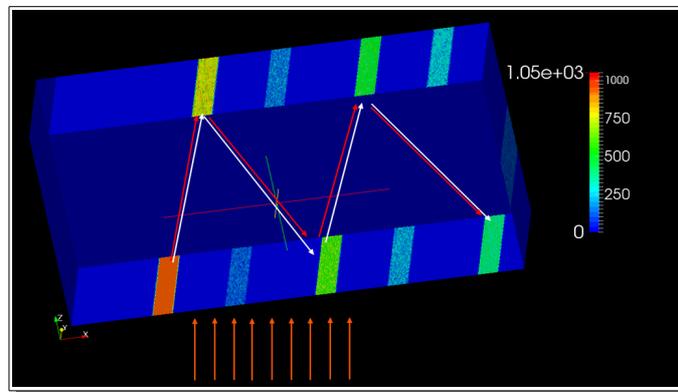
AUTAN est un outil modélisant les phénomènes de rayonnement sur des surfaces semi-transparentes. **AUTAN** calcule pour chaque triangle (que constitue l'objet) le pourcentage de flux solaire réfléchi plus le pourcentage de flux reçu de chacun des autres triangles

fig. 8: température volumique et surfacique d'un Ballon MIR



PAPAGAYO est un outil de modélisation qui prend en entrée les sorties de **AUTAN** permettant de calculer à un instant « t » la température des parois de l'objet que l'on étudie. Les flux issus d' **AUTAN** sont intégrés dans les calculs de **PAPAGAYO**.

fig. 9: les multi-réflexions énergétiques sur un objet



2. ParaVIEW et Cassandra

ParaView est un logiciel libre de visualisation de données.

ParaView (Parallel Visualization Application) est un outil multiplate-forme basé sur VTK permettant la visualisation interactive de gros volumes de données scientifiques 2D et 3D.

L'utilisateur construit interactivement le pipeline graphique (succession de filtres agissant sur les données) de son choix.

L'interface est ergonomique et elle est facile à prendre en main. Il suffit d'apprendre progressivement à connaître les actions des filtres sur les données (possibilité d'en ajouter de nouveaux).

Ce logiciel permet de visualiser des données provenant de domaines très variés : écoulements, images médicales, molécules,...

X. Réalisation

1. Explication et installation de la bibliothèque VTK

Qu'est ce VTK?

Le Visualization ToolKit (ou VTK) est une Bibliothèque « Open Source » de classes C++ qui permet de visualiser des données 2D/3D et de traiter les images.

Elle repose sur le standard OpenGL et elle est intégrable avec des bibliothèques d'interfaces graphiques (Qt, WxWidgets, Java, X11linux, Windows, Mac OS)

Les domaines d'utilisation sont varié :

1. Analyse d'image
2. Imagerie médicale
3. Mécanique des fluides
4. Finance
5. Acoustique
6. Mécanique
7. Éléments finis
8. Géophysique
9. Stéréolithographie
10. Biologie Moléculaire
11. Reconstructions de volumes
12.

Dans notre cas nous utiliserons VTK pour visualiser des objets 3D et nous nous servirons des classes élémentaires et indispensables de la bibliothèque VTK :

- vtkRenderWindow : gère la (ou les) fenêtre(s) d'affichage.
- vtkRenderer : coordonne le rendu en utilisant les sources de lumières, les caméras et bien sûr les acteurs.
- vtkActor : définit un acteur de la scène.
- vtkProperty : définit les propriétés d'apparences d'un acteur : couleur, transparence, comportement par rapport aux lumières.
- vtkMapper : définit la représentation géométrique d'un ou de plusieurs acteurs.

Avant d'aller plus loin :

Comment installer et utiliser VTK avec JAVA dans Eclipse ?

A noter, sur mon poste de travail chez R.Tech, je suis sur une distribution Linux **Scientific Linux (SL) version 7.2 (64bits)** et il est nécessaire d'effectuer l'installation de VTK en ligne de commande.

ETAPE n° 1

On doit télécharger VTK sur <http://www.vtk.org/> (site officiel) et compiler VTK en incluant le JAVA Wrapping (l'enrobage de JAVA)

```
[didierh@silver VTK-7.0.0]$ sh cmake-3.5.2-Linux-x86_64.sh // Installation du compilateur
```

```
//installation des packages nécessaires libXt, Java Development Kit ...
```

```
[didierh@silver VTK-7.0.0]$ rpm -i libXt-devel-1.1.4-6.1.el7.x86_64.rpm
```

```
[didierh@silver VTK-7.0.0]$ rpm -i jdk-8u91-linux-x64.rpm
```

```
....
```

```
// décompression d'eclipse
```

```
[didierh@silver VTK-7.0.0]$ tar xvfz eclipse-java-mars-2-linux-gtk-x86_64.tar.gz
```

```
// décompression de VTK
```

```
[didierh@silver VTK-7.0.0]$ tar xvfz VTK-7.0.0.tar.gz
```

```
//lancement de cmake pour paramétrer la compilation de VTK
```

```
[didierh@silver VTK-7.0.0]$ ccmake .
```

ETAPE n° 2

On doit spécifier dans « ccmake » les paramètres suivants avant de lancer la configuration :

```
BUILD_SHARED_LIBS ON
VTK_USE_RENDERING ON
VTK_WRAP_JAVA ON
```

fig. 12: copie d'écran de « ccmake »

```
didierh@silver:~/Software/VTK-7.0.0
File Edit View Search Terminal Help
Page 1 of 1
BUILD_DOCUMENTATION OFF
BUILD_EXAMPLES OFF
BUILD_SHARED_LIBS ON
BUILD_TESTING OFF
BUILD_USER_DEFINED_LIBS OFF
CMAKE_BACKWARDS_COMPATIBILITY 2.4
CMAKE_BUILD_TYPE Debug
CMAKE_INSTALL_PREFIX /usr/local
DOXYGEN_EXECUTABLE DOXYGEN_EXECUTABLE-NOTFOUND
EXECUTABLE_OUTPUT_PATH /home/didierh
LIBRARY_OUTPUT_PATH
VTK_ANDROID_BUILD OFF
VTK_EGL_DEVICE_INDEX 0
VTK_EXTRA_COMPILER_WARNINGS OFF
VTK_GLEXT_FILE /home/didierh/Software/VTK-7.0.0/Utilities/Parse0GLEx
VTK_GLEXT_FILE /home/didierh/Software/VTK-7.0.0/Utilities/Parse0GLEx
VTK_Group_Imaging OFF
VTK_Group_MPI OFF
VTK_Group_Qt OFF
VTK_Group_Rendering ON
VTK_Group_StandAlone ON
VTK_Group_Tk OFF
VTK_Group_Views OFF
VTK_Group_Web OFF
VTK_IOS_BUILD OFF
VTK_JAVA_INSTALL OFF
VTK_JAVA_JOGL_COMPONENT OFF
VTK_JAVA_SWT_COMPONENT OFF
VTK_PYTHON_VERSION 2
VTK_RENDERING_BACKEND OpenGL2
VTK_SMP_IMPLEMENTATION_TYPE Sequential
VTK_USE_CXX11_FEATURES OFF
VTK_USE_LARGE_DATA OFF
VTK_WGLEXT_FILE /home/didierh/Software/VTK-7.0.0/Utilities/Parse0GLEx
VTK_WRAP_JAVA ON
VTK_WRAP_PYTHON OFF
VTK_WRAP_TCL OFF

BUILD DOCUMENTATION: Build the VTK documentation
Press [enter] to edit option
Press [c] to configure
Press [h] for help
Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
CMake Version 3.5.2
```

ETAPE n° 3

L'étape suivante est de lancer la compilation et d'exécuter le programme qui va générer le JAVA Wrapping de VTK

```
[didierh@silver VTK-7.0.0]$ make  
[didierh@silver VTK-7.0.0]$ make install
```

ETAPE n° 4

Paramétrage de linux, spécification des alias et des variables d'environnement dans le fichier
~/.bashrc

Ajout des chemins

```
« /home/didierh/Software/cmake-3.5.2-Linux-x86_64/bin » dans PATH et de  
« /home/didierh/Software/VTK-7.0.0/lib/ » .....dans LD_LIBRARY_PATH
```

NB : Lorsque vous tapez une commande, le système cherche la commande dans les dossiers spécifiés par la variable PATH, dans l'ordre où ils sont indiqués. Et LD_LIBRARY_PATH est la liste des dossiers où le système doit chercher les bibliothèques d'exécution.

Une fois que les précédentes **Étapes** ont été réalisées, nous pouvons utiliser les bibliothèques .java de VTK que l'on peut copier dans notre projet. A savoir malgré tout que ces bibliothèques de VTK sont indissociables des fichiers .so ou DLL (suivant l'environnement système).



Pour les bibliothèques JAVA, le Garbage Collector (le ramasse miette) s'occupe d'optimiser la mémoire, notamment en libérant la mémoire des objets qui ne sont plus utilisés.

Les objets VTK qui ont été codés en C++ ne sont pas gérés par la JVM et il faut s'assurer quand on n'a plus besoin d'eux et de les détruire afin d'éviter une fuite de mémoire.

liens annexes :

https://fr.wikibooks.org/wiki/VTK/Exemple_introductif#Le_conditionneur

<http://www.vtk.org/doc/nightly/html/classes.html>

<http://www.bu.edu/tech/support/research/training-consulting/online-tutorials/vtk/>

<http://www.bu.edu/tech/support/research/training-consulting/online-tutorials/paraview/>

```

import vtk.*
public class TestVTK {
    static {
        System.loadLibrary("jvawt");
        if (!vtkNativeLibrary.LoadAllNativeLibraries()) {
            for (vtkNativeLibrary lib : vtkNativeLibrary.values()) {
                if (!lib.IsLoaded()) {
                    System.out.println(lib.GetLibraryName() + " not loaded"); }
            }
        }
        vtkNativeLibrary.DisableOutputWindow(null);
    }
    public TestVTK() {}
    public static void main(String[] args) {

        vtkSphereSource sphereSource = new vtkSphereSource();
        sphereSource.SetCenter(0.0, 0.0, 0.0);
        sphereSource.SetRadius(1.0);
        sphereSource.Update();

        vtkPolyData polydata = sphereSource.GetOutput();

        vtkPolyDataMapper polydata_mapper = new vtkPolyDataMapper();
        polydata_mapper.SetInputData(polydata);

        vtkActor actor = new vtkActor();
        actor.SetMapper(polydata_mapper);

        actor.GetProperty().SetColor(1, 0, 0);
        actor.GetProperty().SetOpacity(0.8);
        actor.GetProperty().SetRepresentationToSurface();
        actor.GetProperty().EdgeVisibilityOn();
        actor.GetProperty().SetEdgeColor(1, 0, 0);

        vtkRenderer renderer = new vtkRenderer();
        renderer.AddActor(actor);

        vtkPolyDataNormals normals = new vtkPolyDataNormals();
        normals.SetInputData(polydata); // Passage direct du polydata
        normals.Update();

        vtkArrowSource arrow = new vtkArrowSource();
        arrow.Update();

        vtkGlyph3D glyph3D = new vtkGlyph3D();
        glyph3D.SetInputData(normals.GetOutput()); // Passage direct du polydata normals !
        glyph3D.SetSourceConnection(arrow.GetOutputPort());
        glyph3D.OrientOn();
        glyph3D.SetVectorModeToUseNormal(); // On dit au glyph quoi afficher
        glyph3D.SetScaleFactor(1);
        glyph3D.Update();

        vtkPolyDataMapper normals_polydata_mapper = new vtkPolyDataMapper();
        normals_polydata_mapper.SetInputConnection(glyph3D.GetOutputPort());

        vtkActor actor_normals = new vtkActor();
        actor_normals.SetMapper(normals_polydata_mapper);
        actor_normals.GetProperty().SetColor(0, 1, 0);

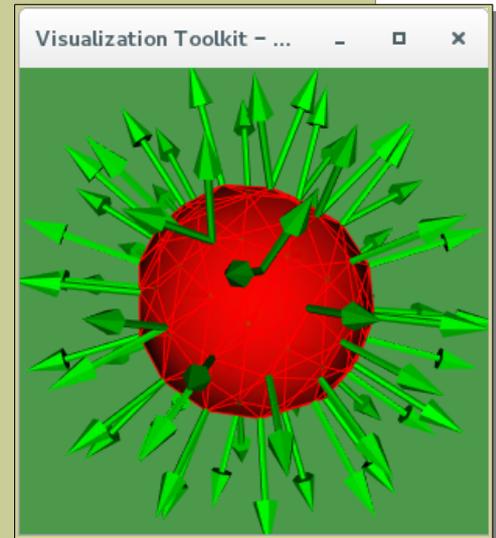
        renderer.AddActor(actor_normals);

        vtkRenderWindow renderWindow = new vtkRenderWindow();
        renderWindow.AddRenderer(renderer);
        vtkRenderWindowInteractor renderWindowInteractor = new vtkRenderWindowInteractor();
        renderWindowInteractor.SetRenderWindow(renderWindow);

        renderer.SetBackground(.3, .6, .3);

        renderWindow.Render();
        renderWindowInteractor.Start();
    }
}

```



Affichage d'une sphère avec ses bords + transparence

```
import vtk.vtkActor;
import vtk.vtkNativeLibrary;
import vtk.vtkPolyDataMapper;
import vtk.vtkRenderWindow;
import vtk.vtkRenderWindowInteractor;
import vtk.vtkRenderer;
import vtk.vtkSphereSource;

public class Exemple {

    static {
        System.loadLibrary("jvmt");
        if (!vtkNativeLibrary.LoadAllNativeLibraries()) {
            for (vtkNativeLibrary lib : vtkNativeLibrary.values()) {
                if (!lib.IsLoaded()) {
                    System.out.println(lib.GetLibraryName() + " not loaded");
                }
            }
        }
        vtkNativeLibrary.DisableOutputWindow(null);
    }

    public Exemple() {
        // TODO Auto-generated constructor stub
    }

    public static void main(String[] args) {

        // Instanciation de l'objet sphère et attributs
        vtkSphereSource sphere = new vtkSphereSource();
        sphere.SetRadius(2.0);
        sphere.SetThetaResolution(16);
        sphere.SetPhiResolution(12);
        sphere.SetCenter(0.0, 0.0, 0.0);

        sphere.Update();

        // Mapping de la bibliothèque graphique
        vtkPolyDataMapper map = new vtkPolyDataMapper();
        map.SetInputData(sphere.GetOutput());

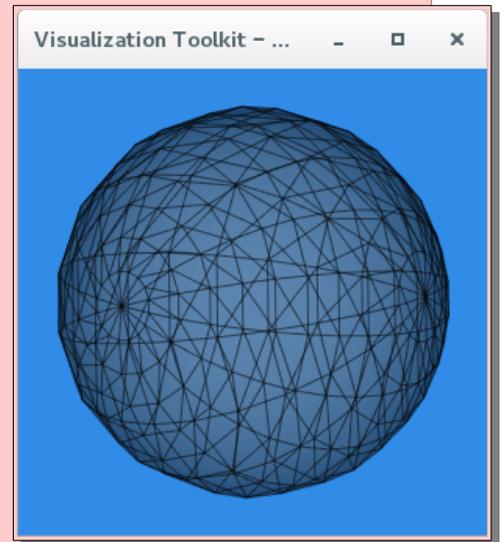
        // Instanciation d'un acteur et attributs
        vtkActor aSphere = new vtkActor();
        aSphere.SetMapper(map);
        aSphere.GetProperty().SetColor(128/255., 128/255., 128/255.);
        aSphere.GetProperty().SetOpacity(0.32); // Transparence 32%
        aSphere.GetProperty().SetEdgeVisibility(1); // Affichage des bords
        aSphere.GetProperty().SetRepresentationToSurface();
        // Présentation surfacique de l'objet

        // Création de la fenêtre de rendu
        vtkRenderWindow renWin = new vtkRenderWindow();
        vtkRenderer ren1 = new vtkRenderer();
        renWin.AddRenderer(ren1);

        // Instanciation de l'interacteur
        vtkRenderWindowInteractor iren = new vtkRenderWindowInteractor();
        iren.SetRenderWindow(renWin);

        // Ajout de l'objet sphère sur un fond bleu
        ren1.AddActor(aSphere);
        ren1.SetBackground(49/255., 140/255., 231/255.);

        // Rendu de l'image
        renWin.Render();
        iren.Start();
    }
}
```

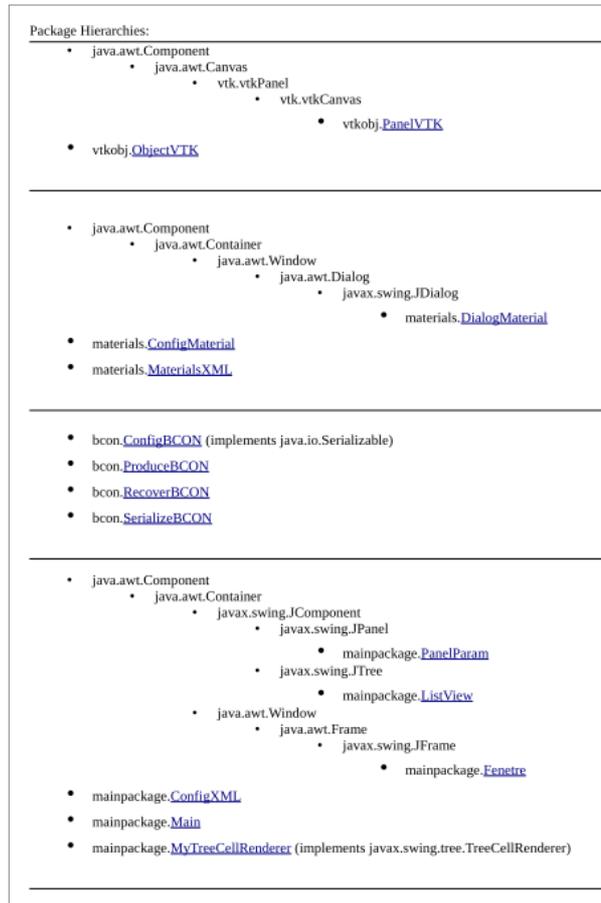
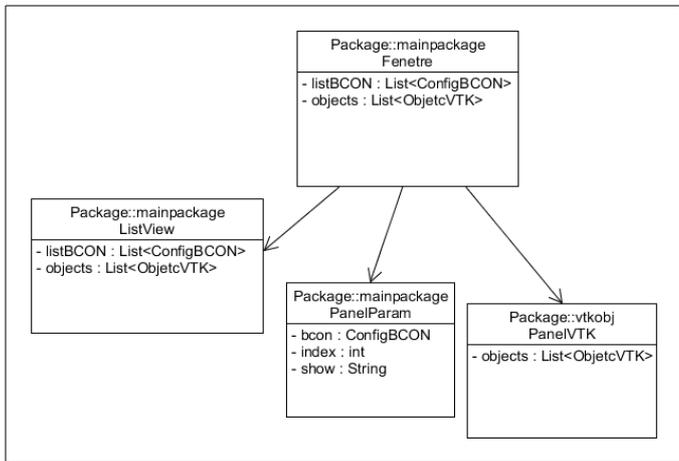


2. Réalisation de la JavaDoc

Pour faciliter le suivi et la maintenabilité de l'outil , la JavaDoc a été renseignée.

3. Description des Packages et Classes

fig. 13: packages et Classes de Mistral-RAD



On a créé trois JavaBEAN qui sert de support de données :

- ✓ pour la gestion des sous-ensembles géométriques ou « patch » [ConfigBCON],
- ✓ pour la gestion des matériaux [ConfigMaterial] et
- ✓ pour la gestion des objets 3D (VTK) [ObjectVTK]

Structurellement on a une classe principale « Fenetre » qui sert d'enveloppe à notre interface graphique et de contenant aux trois sous-ensembles que sont :

- ✓ l'arborescence (« ListView »),
- ✓ le panneau de saisie (« PanelParam ») et
- ✓ et le volet de visualisation 3D (« PanelVTK »)

La fenêtre principale gère une liste (ArrayList) de type « ConfigBCON » et de type « ObjectVTK ».

L'arborescence gère deux listes de type « ConfigBCON » et de type « ObjectVTK ».

Le panneau de saisie « PanelParam » gère un seul élément de la liste de type « ConfigBCON » (dont il connaît l'index dans la liste).

Le panneau de visualisation 3D « PanelVTK » gère une liste de type « ObjectVTK ».

(voir Annexe Diagramme de Classes)

La Classe « ConfigBCON » est 'serializable' ce qui permet facilement la sauvegarde des données et leur restauration.

Les relations entre les classes de l'interface graphique et les données (JavaBEAN) se font via une passage par référence des données à la classe enfant (cible).

Le clic de la souris ou le changement d'état des composants graphiques (listview textfield combobox button) génèrent des événements et ordonnent le bon fonctionnement entre les divers éléments de l'interface graphique : arborescence / panneau de saisie / visualisation d'objet 3D.

4. Exemples de Code et appels de fonctions

Restauration des données sauvegardées (dans la Classe Fenetre)

commentaires

```
SerializeBCON importBCON = new SerializeBCON();
List<ConfigBCON> tampon = null;
tampon = importBCON.OpenBCON(mframe);

if(tampon != null && tampon.size() >0)
{

    listBCON = tampon;
    mtree.initListView(listBCON);

    makeObjectsVTK(listBCON);

    mtree.setObjects(objects);

    panelVTK.initPanel(objects);

    panelVTK.setClickRight(true);
    panelVTK.resetCamera();
}
if(isEmpty()) {
    JOptionPane.showMessageDialog(mframe,
        "Error no OBJECT !", "Message GENERER",
        JOptionPane.ERROR_MESSAGE);
    return ;
}
mtree.setSelectionRow(0);

param.setIndex(-1);

param.show("0");
}
```

initialise un Objet SerializeBCON
crée une ArrayList de type ConfigBCON
récupère des données sauvegardées

condition
s'assurer que la liste tampon n'est pas vide

initialise l'arborescence avec la ArrayList

init. et généré la liste d'objets VTK
« objects »
qui doivent être afficher dans le panneau de visualisat*3D

init. « objects » dans l'arborescence

init. liste d'objets VTK dans le panneau 3D

autorise le clic-droit dans le panneau 3D
initialise la camera dans le panneau 3D

condition
si « ListBCON » est vide on affiche un message et on sort

sélectionne la racine de l'arborescence

indique qu'aucun objet est sélectionné
dans le volet de saisie

affiche un panneau vide
dans le volet de saisie

Changement de la couleur d'un objet 3D quand on clique sur le champ « textColor » (JtextField) dans la classe PanelParam (panneaux de saisie)

commentaires

<pre> Color color = null; if(bcon.getColor() !=null) color = bcon.getColor(); color = JcolorChooser.showDialog(panel, "Choice one COLOR", color); if(color != null && textColor.getBackground() != color) { textColor.setBackground(color); saveConfigBCON(); object.setColor(color); ((Fenetre) mframe).initVTK(); mtree.initVisualization(); mtree.repaint(); } </pre>	<p>crée une variable « color » initialise « color »</p> <p>affiche la boîte de dialogue JcolorChooser</p> <p>CONDITION si « color » est différent du 'background' de « textColor ou non NULL</p> <p>affecte la nouvelle couleur au 'background' de « textColor »</p> <p>sauvegarde les valeurs du panneau de saisie</p> <p>change la couleur de l'objet VTK initialise le panneau 3D initialise l'arborescence force la refonte de l'affichage de l'arborescence</p>
---	--

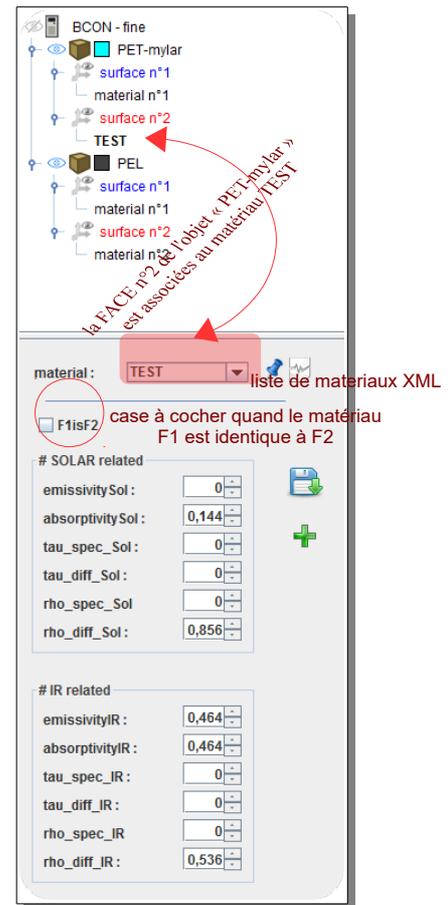
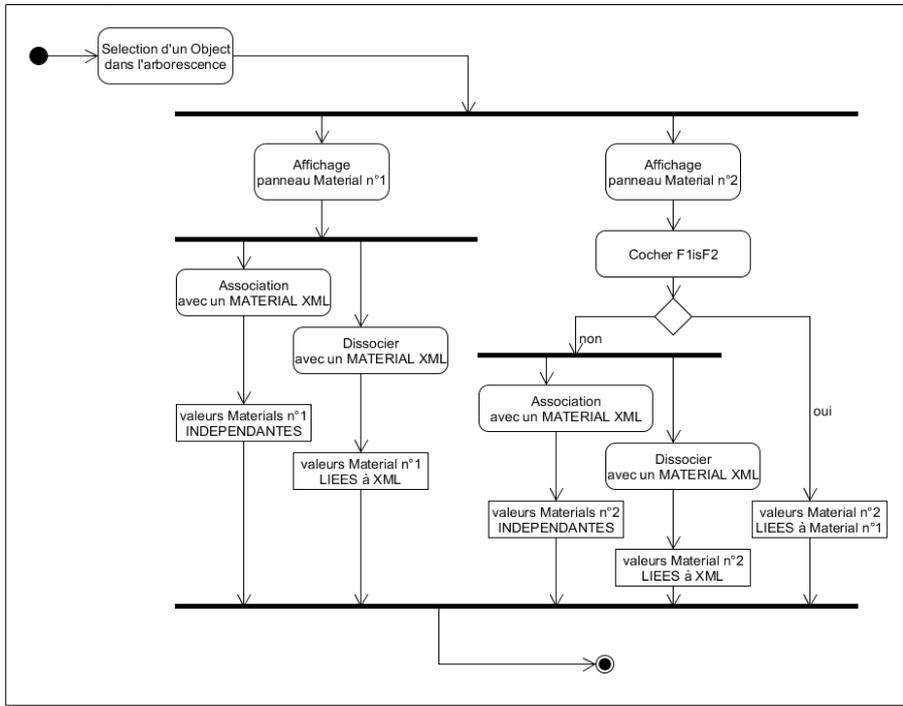
Dans l'arborescence, quand on clique sur un élément de type « patch » (de niveau 1) dans la classe Fenetre (voir schéma)

commentaires

<pre> DefaultMutableTreeNode node = (DefaultMutableTreeNode) int level = node.getLevel(); if(level==0 listBCON == null listBCON.size() < 1) { param.show("0"); return; } if(level==1) { int index = node.getRoot().getIndex(node); if(index!=param.getIndex()) { param.setIndex(index); param.setBcon(listBCON.get(index)); param.setObject(objects.get(index)); param.initConfigBCON(); } param.show("1"); } </pre>	<p>récupère le nœud de la sélection</p> <p>récupère le niveau de la sélection</p> <p>condition si le niveau est 0 ou la listBCON est vide : on sort et affiche un panneau de saisie vide</p> <p>condition le niveau == 1 récupéré l'index de l'objet dans la liste « listBCON »</p> <p>CONDITION si l'index dans l'arborescence est différent de l'index du panneau de saisie</p> <p>affecte le nouveau index au panneau de saisie</p> <p>affecte un élément ConfigBCON dans le panneau de saisie de « listBCON »</p> <p>affecte un élément ObjectVTK dans le panneau de saisie de la liste objects</p> <p>initialise les champs du panneau</p> <p>affiche le panneau correspondant au niveau ① dans l'arborescence</p>
---	--

5. Gestion des Matériaux dans le panneau de saisie

fig. 14: gestion de l'association des Matériaux de la base de données XML dans les panneaux de saisie



Un patch dispose de deux FACES et nous pouvons affecter un matériau différent sur les deux FACES. Dans le cas où les deux FACES ont un matériau identique, on coche la case « F1isF2 » dans le panneau de saisie pour l'élément « material n°2 » de l'arborescence.

Pour chaque FACE on peut associer un matériau de la base de données XML. Pour cela il faut choisir un élément dans la liste déroulante et de cliquer sur le bouton 

Pour les valeurs de saisie « #SOLAR related » ou « #IR related », leurs libellés deviennent rouge si elles sont erronées ou non conformes. Après avoir associé un matériau à une FACE, si les valeurs du matériau ne sont pas identiques à la base de données XML, un astérisque apparaît devant le libellé.

Code Java dans le panneau de saisie quand la case « F1isF2 » est coché dans la classe PanelParam

commentaires

```
public void show(String name) {  
  
    show = name;  
  
    if(name == "4" || name == "5" ) {  
  
        updateListMaterials();  
  
        if(name == "4") {  
            changeMaterial(1);  
            checkFis.setVisible(false);  
            setVisiblePanelMaterial(true);  
        }  
  
        if(name == "5") {  
            changeMaterial(2);  
            checkFis.setVisible(true);  
            setVisiblePanelMaterial(!checkFis.isSelected());  
  
        }  
  
        initPanelMaterial();  
        cardLayout.show(panel, "4");  
        return;  
    }  
  
    cardLayout.show(panel, name);  
}
```

fonction permettant d'afficher un panneau de saisie.

A savoir : 'panel' est le contenant des cardLayout

indique à la variable 'show' le nom du panneau qui sera affiché

Cas particulier gestion des panneaux
« material n°1 » eq name == '4' et
« material n°2 » eq name == '5'

Met à jour la liste du menu déroulant

si on affiche le panneau « material n°1 »
..met à jour les valeurs du matériau
..rend invisible « F1isF2 »
..enabled les champs de saisie

si on affiche le panneau « material n°2 »
..met à jour les valeurs du matériau
..affiche la case à cocher « F1isF2 »
..enabled les champs de saisie

si « F1isF2 » n'est pas cochée
sinon on n'active pas les champs

A noter si « F1isF2 » est cochée
les valeurs du matériau de la FACE 2 sont
identiques à celles de la FACE 1

ré-initialise les composants de saisie
affiche le panneau de saisie des Matériaux
on sort de la fonction

affiche les autres panneaux
si name!= '4' et si name!= '5'

Code Java dans le panneau de saisie quand on utilise le bouton d'« Association » (voir Guide de l'utilisateur)

```
1)  if(((String) cbMaterial1.getSelectedItem()).equals("...")) return;
2)  if(show == "4") {
3)
4)
5)      bcon.setCompareF1(((String) cbMaterial1.getSelectedItem()));
6)
7)      bcon.setMaterial1(new MaterialsXML().getMaterial(
8)                          (String)cbMaterial1.getSelectedItem());
9)
10)     mtree.changeValue1((index) *5 + 1 + 2, (String) cbMaterial1.getSelectedItem());
11)
12)     if(checkFis.isSelected()) {
13)         mtree.changeValue1((index) *5 + 1 + 4, (String)
14)                             cbMaterial1.getSelectedItem());
15)     }
16) } else {
17)     bcon.setCompareF2(((String) cbMaterial1.getSelectedItem()));
18)
19)     bcon.setMaterial2(new MaterialsXML().getMaterial(
20)                         (String)cbMaterial1.getSelectedItem());
21)
22)     mtree.changeValue1((index) *5 + 1 + 4, (String) cbMaterial1.getSelectedItem());
23) }
24)
25) initConfigBCON();
26)
27) initPanelMaterial();
```

Dessous les commentaires du code ci-dessus : (faire le lien avec les numéros de ligne)

Remarques : '**bcon**' est l'objet issu de la classe « Fenetre » qui a été passé par référence à la classe « PanelParam » qui correspond aux valeurs intrinsèques de l'objet ou « patch » sélectionné dans l'arborescence.

```
1)  on sort de la fonction si aucun élément de la liste est sélectionné
2)  si c'est le panneau « Material n°1 » qui est affiche
3)
4)      on affecte le nom sélectionné de la liste de Matériaux à « CompareF1 » de 'bcon'
5)
6)      on affecte les valeurs du Matériaux de la base de données XML à 'bcon'
7)
8)      on donne le nom du Matériau dans l'arborescence à la place de « material n°1 »
9)
10)
11)     si la case « F1isF2 » est coché on change aussi la valeur de « material n°2 »
12)     dans l'arborescence
13)
14)
15) si c'est le panneau « Material n°2 » qui est affiche
16)
17)     on affecte le nom sélectionné dans la liste de Matériaux à « CompareF2 » de 'bcon'
18)
19)     on affecte les valeurs du Matériaux de la base de données XML à 'bcon'
20)
21)     on donne le nom du Matériau dans l'arborescence à la place de « material n°1 »
22)
23)
24)
25) Initialise les valeurs des composants des PANELS avec celle du champ 'bcon'
26)
27) ré-initialise les composants de saisie
28)
```

6. Déploiement de l'application

Pendant mon stage j'ai utilisé **Git en ligne de commande** qui m'a permis de déposer mon code pour qu'il puisse être partagé et sauvegardé sur un serveur local.

Pour pouvoir être utilisé, mon application doit être publiée et être installée sur le poste du client, on doit pour cela utiliser un installateur.

Chez R.Tech pour les applications Java on utilise IzPack

voir lien :

<http://izpack.org/downloads/>

<https://izpack.atlassian.net/wiki/display/IZPACK/Sample+Install+Definition>

Ci-dessous les étapes pour générer l'exécutable .jar pour installer Mistral-RAD (sous Windows)

1) on télécharge IzPack sur le site du logiciel <http://izpack.org/downloads/>

2) on exécute izpack-dist-5.0.8.jar en double cliquant dessus pour installer le package soit on utilise la commande « compile » de IzPack en local ou sinon on copie le répertoire « C:\Program Files\IzPack\bin » pour lancer la commande à distance ou sur une autre poste

3) On rassemble tous les fichiers nécessaires à notre application dans un répertoire : image, fichier .jar , librairies, ainsi que les fichiers .bat ou .sh pour lancer notre application quand elle sera installée sur le poste hôte.

4) on doit éditer le fichier install.xml, les panels et autres ressources nécessaires au paramétrage ou construction de notre installateur :

install.xml / ProcessPanel.xml / shortcutSpec.xml / welcome.html / logo_RTECH.png

- le fichier welcome.html correspond à la page de garde de notre installateur (contenant le nom de l'application, le n° de version et le nom de l'entreprise). Page qui sera affichée en deuxième position.
- le fichier shortcutSpec.xml correspond à la configuration des raccourcis.
- le fichier install.xml est le fichier coordonnateur des différents panneaux , qui définit les ressources et les fichiers ou répertoire qui vont être copiés dans le répertoire d'installation (voir en annexe le fichier).

5) On lance la commande suivante pour compiler et créer notre fichier install.jar :

```
[didierh@silver ant ...]$ cd C:\Users\Utilisateur\Desktop\softwarepassion\  
[didierh@silver ant ...]$ "C:\Program Files\IzPack\bin" compile install.xml -b . -o install.jar -k standard
```

ci dessous le déploiement de Mistral-RAD sur un poste Windows en utilisant le fichier « install.jar » précédemment créé (en double cliquant sur le fichier install.jar) .

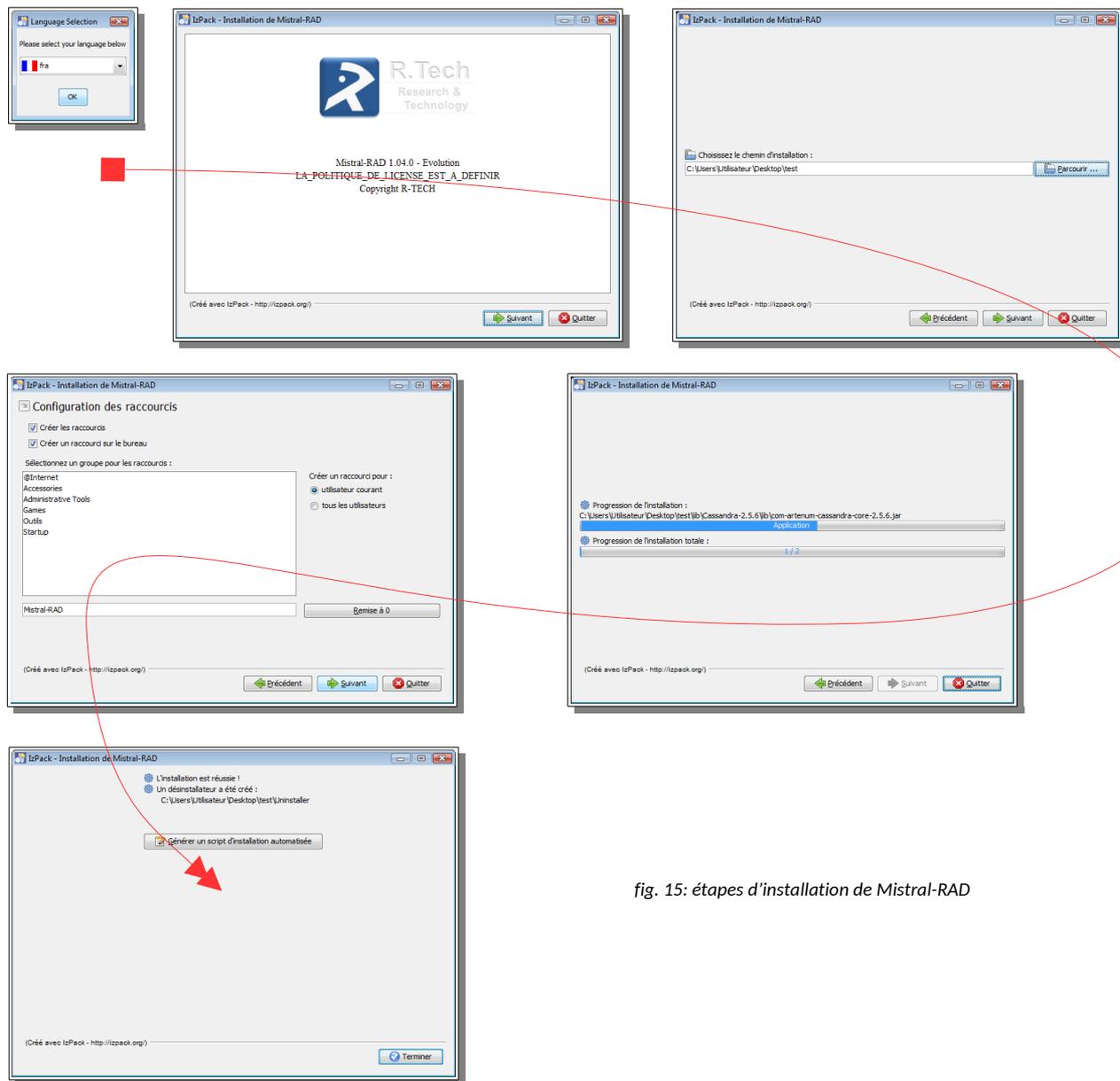


fig. 15: étapes d'installation de Mistral-RAD

Ci-dessous le détail des fichiers qui lancent l'application :
script Batch Windows (fichier .bat)

```

echo off
set CLASSPATH=lib\Cassandra-2.5.6\dependencies\thirdparty\vtk-5.8.0-win7-32b
set PATH=%CLASSPATH%;%PATH%

java -cp "lib/mistral-rad.jar;lib\Cassandra-2.5.6\dependencies\main\vtk-5.8.0.jar;lib/jdom-2.0.6/jdom-2.0.6.jar"
mainpackage.Main

```

ou script Shell Unix (fichier .sh)

```

#!/bin/sh
export LD_LIBRARY_PATH= ./lib/Cassandra-2.5.6/dependencies/thirdparty/vtk-5.8.0-win7-32b:$LD_LIBRARY_PATH

java -classpath ".lib/mistral-rad.jar;./lib/Cassandra-2.5.6/dependencies/main/vtk-5.8.0.jar;./lib/jdom-2.0.6/jdom-2.0.6.jar" mainpackage.Main

```

Petite parenthèse concernant ANT

Au sein de l'entreprise R.Tech, pour les applications JAVA on utilise ANT. ANT est un outil qui permet d'automatiser des opérations répétitives tout au long du cycle de développement logiciel telles que la compilation, la génération de documents (Javadoc) ou l'archivage au format JAR .

ANT fournit par défaut un panel de fonctionnalités, qui auront le même comportement sur tous les systèmes.

La définition des tâches à exécuter (écrite en Java) se fait via un fichier de configuration XML. Le fichier de configuration XML se nomme build.xml et se trouve habituellement à la racine du projet.

Pour exécuter une tâche, il suffit de taper une ligne de commande comme ci-dessous :

```
[didierh@silver ant ...]$ ant [options] [cible]
[didierh@silver ant ...]$ ant maTache
```

(la tâche « maTache » est défini dans le fichier .xml)

Ci-dessous la liste des tâches qui sont possibles :

Archive / Audit/Coverage / Compile / Deployment / Documentation / Execution / File / Miscellaneous / Testing

voir le lien suivant : <http://ant.apache.org/manual/taskoverview.html>

ANT permet par exemple pendant la nuit de compiler et déployer une application à partir de leur commit (dépôt dans Git). On peut mettre à disposition l'application avec la dernière version du code ou faire des tests de non-régressions ou avertir le développeur si son code n'a pas pu se compiler ...

XI. Améliorations

A savoir que l'ensemble des spécifications demandées ont été réalisées dans un délai court de cinq semaine..

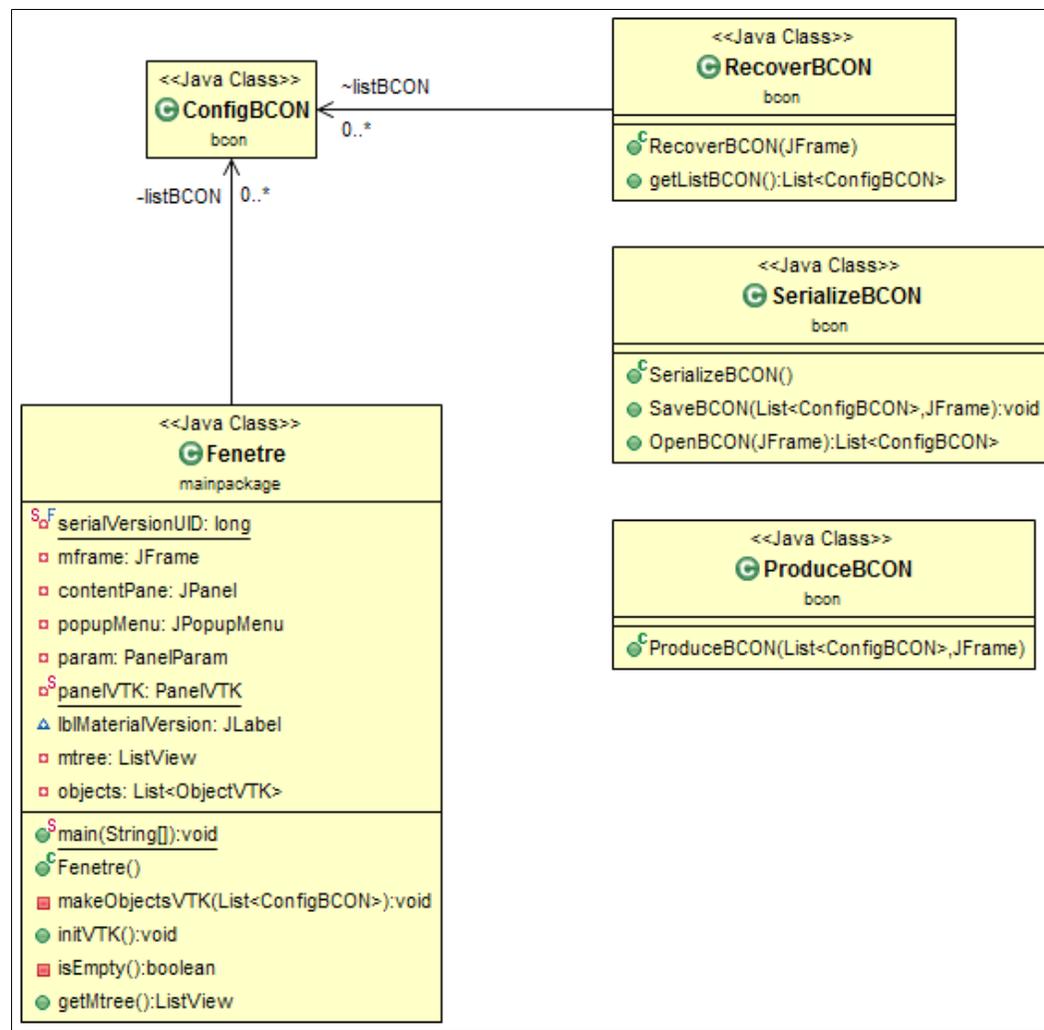
Malgré tout ci-dessous la liste des améliorations qui sont à prévoir :

- (1) Revoir et améliorer la boîte de dialogue de gestion des Matériaux
- (2) Concernant les évolutions les plus intéressantes c'est de pouvoir lancer le calcul dans AUTAN/Papagayo et de pouvoir afficher le résultat.
- (3) Voir et effectuer des correctifs concernant l'affichage du Logo quand on redimensionne la fenêtre pour qu'il puisse rester en haut et à droite de l'écran.
- (4) Concernant l'affichage VTK, éviter dans des cas rares la superposition de VTK au dessus des composants AWT/SWING
- (5) Effectuer des tests sous machine Mac, il se peut qu'un effet de superposition des élément visuel VTK soit flagrant sur les éléments de l'interface graphique AWT/SWING.
- (6) Ajout une gestion de la langue pour l'interface graphique de l'utilisateur avec la librairie Java Internationalization (i18n) Libraries
- (7) Gestion de UNDO dans les champs de saisie pour défaire certaine modification ou changement de valeur
- (8) Amélioration de la visualisation de SURFACES en utilisant une autre solution que l'affichage des normales sous forme de flèches

XIII. Annexe Diagrammes de classe des différents éléments du programme

Éléments / paramètres BCON (conditions limites des différents objets que constitue le fichier final)

fig. 17: diagramme / données de conditions limites



Liens entre les différent éléments de l'interface graphique

fig. 18: diagramme / interface graphique

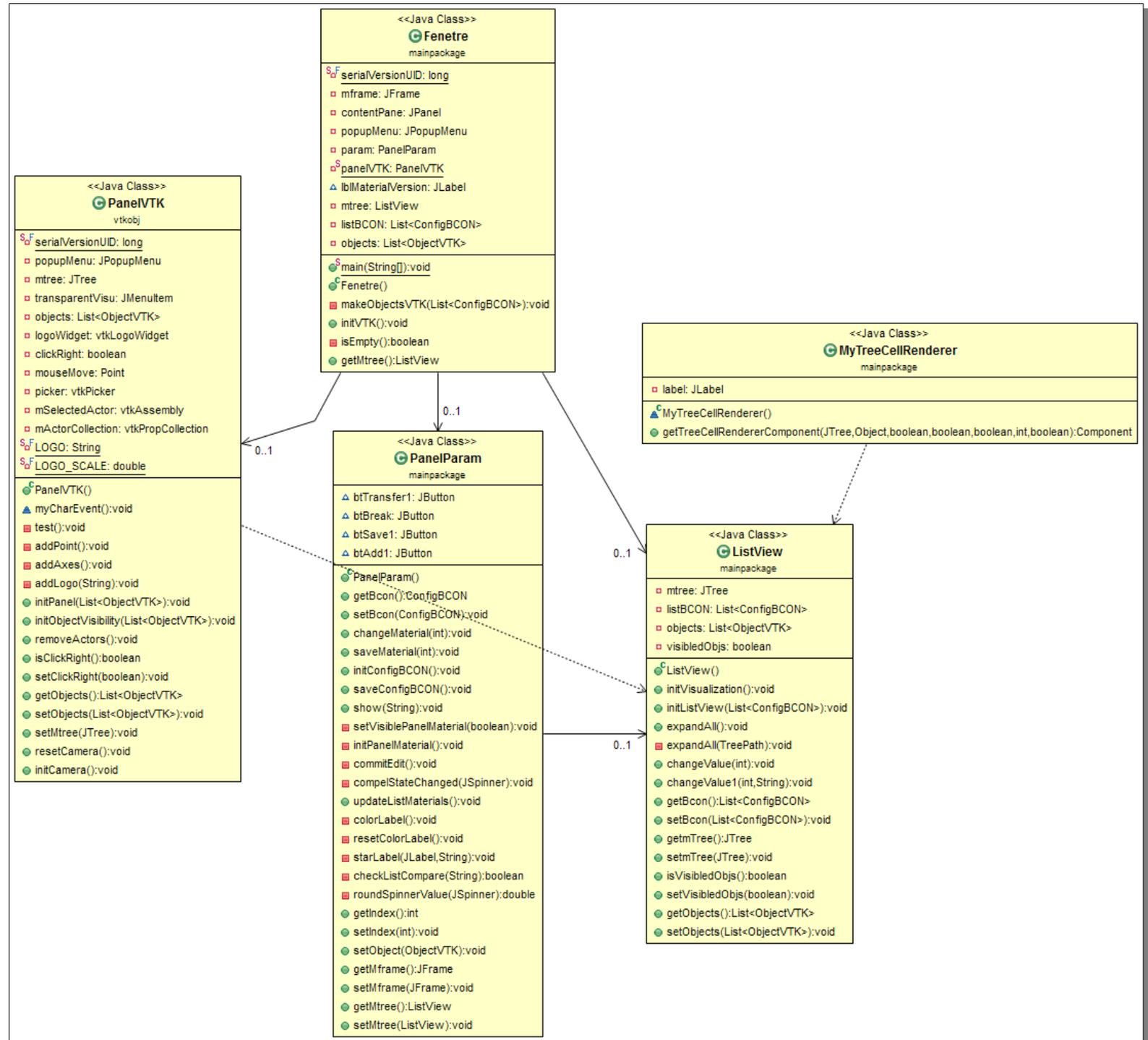


fig. 19: diagramme / éléments VTK

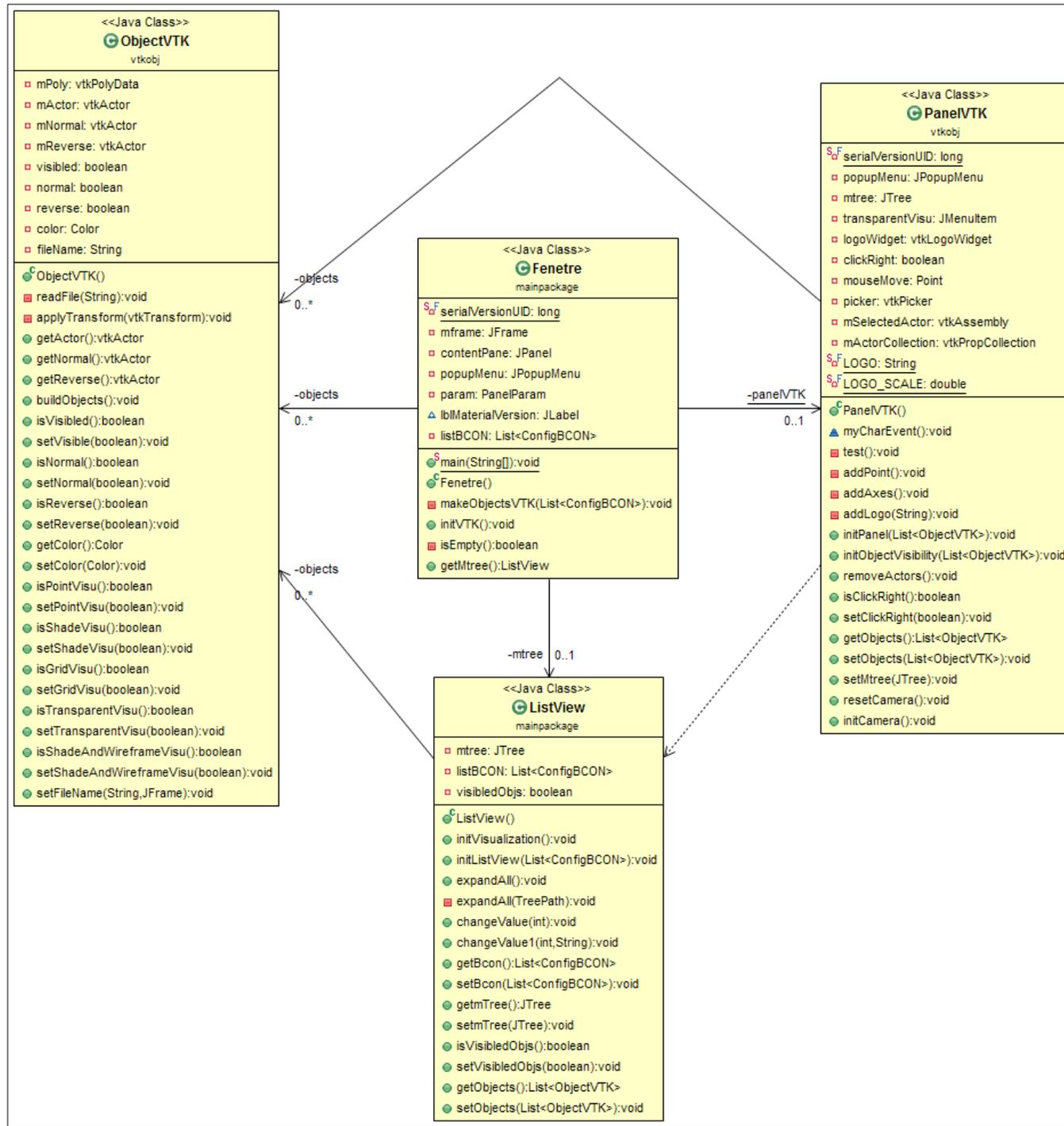
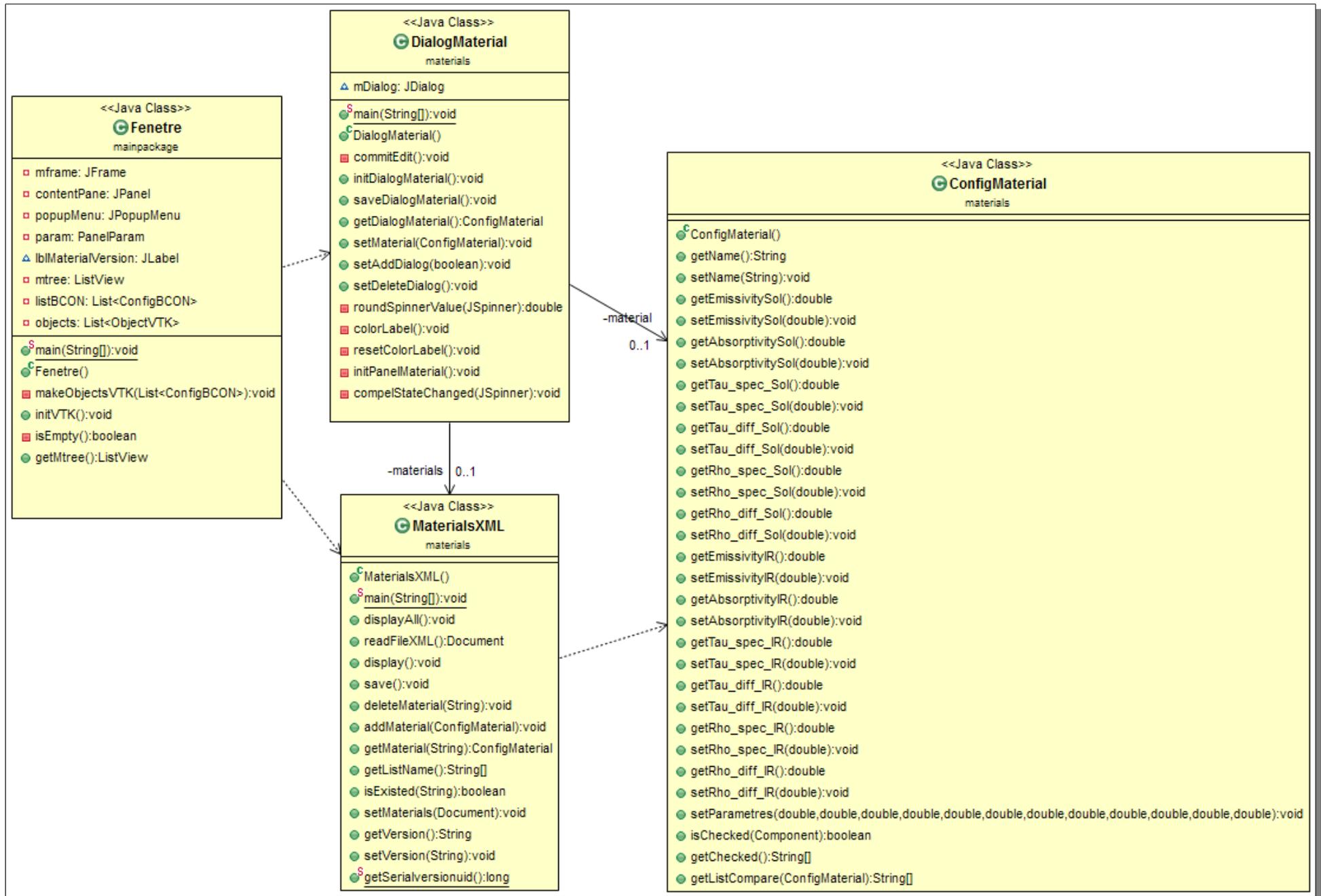


fig. 20: diagramme / base de données des Matériaux



XIV. Annexe Planning des tâches du projets

fig. 21: planning des tâches réalisées

taches	package et classes (projet JAVA)	22/04/16	25/04/16	26/04/16	27/04/16	28/04/16	29/04/16	02/05/16	03/05/16	04/05/16	05/05/16	06/05/16	09/05/16	10/05/16	11/05/16	12/05/16	13/05/16	16/05/16	17/05/16	18/05/16	19/05/16	20/05/16	23/05/16	24/05/16	25/05/16	26/05/16	27/05/16	30/05/16	31/05/16	01/06/16	02/06/16	03/06/16	06/06/16	07/06/16	08/06/16	09/06/16	10/06/16			
Installation de VTK + Eclipse		█	█	█																																				
Réunion et présentation + restitution des attentes		█							█											█																				█
Découpage des actions / Réflexions		█	█	█	█	█	█	█	█	█			█		█							█																		
Familiarisation avec la bibliothèque VTK		█	█	█	█	█	█	█	█	█																														
Visualiser un objet 3D dans une fenêtre	GuiVTK - PanelVTK	█	█	█	█	█	█	█	█	█																														
Lire des objets .obj .vtk .stl	ModelVTK - ObjectVTK	█	█	█	█	█	█	█	█	█																														
Gestion de la liste des objets (Jtable)	ConfigBCON		█	█	█	█	█	█	█	█																														
Creation d'une boîte de dialogue pour les paramètres BCON	DialogBCON		█	█	█	█	█	█	█	█																														
Creation d'onglet pour gérer les FACES																																								
Boîte de dialogue pour selection un fichier																																								
Boîte de dialogue pour changer la couleur																																								
Changer l'apparence et la couleur des objets																																								
Génerer le fichier bcon-fine pour 'AUTAN'	ProduceBCON - RecoverBCON																																							
Récuperer un fichier bcon-fine																																								
Serialization / sauvegarde	SerializeBCON																																							
Gestion fichier XML(Jdom)	ConfigMaterial - DialogMaterial - MaterialXML																																							
Gestion des panneaux de Saisie (CardLayout)	PanelParam																																							
Gestion volet Arborescence (ListView)	ListView																																							
Gestion du volet VTK (objet 3D)	ObjectVTK - PanelVTK																																							

fig. 22: détail des tâches réalisées

Détails de tâches réalisées pendant le STAGE

date	commentaire
2016-04-22	Présentation du projet et installation de VTK sur une machine (test des application ANEMO + SEIS)
2016-04-23	[Travail à la maison / installation de VTK sur une machine Windowd (VISTA)]
2016-04-24	Présentation projet PAPAGAYO + réalisation d'un petit planning + découpage des actions / Création de la classe ConfigBCON + serialization
2016-04-25	
2016-04-26	Installation de ParaView / Gestion de Jtable [A la maison, test de VTK avec la version 5.8 Cassandra (...)]
2016-04-27	Création de la boîte de dialogue BCON (paramètres des conditions limites) / Gestion des item dans Jtable + objets VTK
2016-04-28	Fin de Jtable + Gestion fichier 3D (.vtk .obj .stl) (extension) / Affichage des objets 3D et modification de leur apparence Dans la fenêtre de l'affichage des objets 3D, gestion du bouton droit de la souris (pour le menu Popup qui permet de modifier l'apparence des objets)
2016-04-29	Modification de la couleur des objets / Ajout de la classe ObjectVTK / Gestion des normales et de leur inverse
2016-04-30	
2016-05-01	Intégration normales / Génération du fichier bcon-fine / Gestion des fenêtres Visualisation+Paramètres BCON / Récupération d'un fichier bcon-fine / Gestion des fichiers & messages (exceptions)
2016-05-02	
2016-05-03	Présentation n°1 + Demo / Nouvelle réflexion sur l'interface graphique + gestion des Matériaux dans un fichier XML
2016-05-04	Gestion Matériaux XML (Jdom)
2016-05-05	
2016-05-06	
2016-05-07	
2016-05-08	
2016-05-09	Création d'une nouvelle interface graphique : Jtree (ListView) + VTK
2016-05-10	
2016-05-11	Gestion des icônes et lignes (row) dans Jtree / Tests / Modification de la classe ConfigBCON
2016-05-12	Logo R-TECH / Gestion du clic dans ListView (event) dans le panel VTK
2016-05-13	Creation et gestion des Panels dans la classe panelParam (CardLayout) / (Bug avec panelVTK)
2016-05-14	
2016-05-15	
2016-05-16	Gestion des couleur / modification de la classe ObjetVTK (gestion de l'affichage des vtkactor)
2016-05-17	Gestion des GeometryFile (exception) + Gestion des JComboBox / Gestion des normales
2016-05-18	Bouton ajout/supprimer pour MaterialXML / Gestion MaterialXML dans panelParam
2016-05-19	Présentation n°2 + Demo / Modification du Background (dégrade de couleur) / Inversion dans l'arborescence des lignes Surface ... & Material ... / Ajout d'une couleur pour distinguer les FACES
2016-05-20	Affichage des paramètres MATERIAL quand on sélectionne la ligne "material n°.."
2016-05-21	
2016-05-22	
2016-05-23	Gestion de bug concernant MATERIAL ...
2016-05-24	Ajout d'un numéro de version dans materialXML / Ajout bouton ADD et SAVE pour Material
2016-05-25	Ajout configXML / Modification du nom du Material dans la Listview (arborescence) / Utilisation d'un seul PANEL pour les Matériaux / Fonction de vérification des valeurs de saisie pour les Matériaux
2016-05-26	UML installation d'un plugin (Diagramme de classe) / Gestion de la couleur des Labels / Fonction pour arrondir les valeurs issues des Jspinner / Tests / Ajout de boutons pour rafraîchir la listes des Materials & dissocier un Material
2016-05-27	Modification du menu popupMenu de VTK / Commencement de la JavaDoc / Création des diagrammes de Classes
2016-05-28	
2016-05-29	
2016-05-30	Présentation n°3 + Demo / Réalisation des modifications
2016-05-31	Réalisation Dossier
2016-06-01	Réalisation Dossier
2016-06-02	Réalisation Dossier
2016-06-03	Réalisation Dossier
2016-06-04	Réalisation Dossier
2016-06-05	Réalisation Dossier
2016-06-06	
2016-06-07	
2016-06-08	
2016-06-09	
2016-06-10	Démonstration

```

<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<installation version="1.0">
  <info>
    <appname>Mistral-RAD</appname>
    <appversion>1.04.0 - Evolution</appversion>
    <authors>
      <author name="R.Tech-engineering" email="support@rtech-engineering.com"></author>
    </authors>
    <url>http://www.rtech-engineering.com</url>
    <javaversion>1.8</javaversion>
  </info>
  <variables>
    <variable name="DesktopShortcutCheckboxEnabled" value="true"></variable>
  </variables>
  <!-- The gui preferences indication. Sets the installer window to 640x480. It will not be able to change the size. -->
  <guiprefs width="640" height="480" resizable="no">
  </guiprefs>
  <!-- The locale section. Include the English and French langpacks. -->
  <locale>
    <langpack iso3="fra"/>
  </locale>
  <resources>
    <res src="welcome.html" id="HTMLHelloPanel.info"/>
    <res src="logo_RTECH.png" id="logoRTECH"/>
    <res src="ProcessPanel.xml" id="ProcessPanel.Spec.xml" />
    <res src="shortcutSpec.xml" id="shortcutSpec.xml"/>
    <res src="Unix_shortcutSpec.xml" id="Unix_shortcutSpec.xml"/>
    <res src="logo_RTECH.png" id="logo_R-TECH"/>
  </resources>
  <native type="izpack" name="ShellLink.dll"/>
  <native type="izpack" name="ShellLink_x64.dll"/>
  <!-- The panels section. We indicate here which panels we want to use. The order will be respected. -->
  <panels>
    <panel classname="HTMLHelloPanel"/>
    <panel classname="TargetPanel"/>
    <panel classname="InstallPanel"/>
    <panel classname="ShortcutPanel"/>
    <panel classname="FinishPanel"/>
  </panels>
  <!-- The packs section. We specify here our packs. -->
  <packs>
    <pack name="Application" required="yes" preselected="yes" hidden="no">
      <description>The executable file</description>
      <file src="lib/mistral-rad.jar" override="true" targetdir="$INSTALL_PATH/lib" />
      <file src="lib/Cassandra-2.5.6" override="true" targetdir="$INSTALL_PATH/lib" />
      <file src="lib/jdom-2.0.6" override="true" targetdir="$INSTALL_PATH/lib" />
      <file src="toCopy/config.xml" targetdir="$INSTALL_PATH" />
      <file src="toCopy/materials.xml" targetdir="$INSTALL_PATH" />
      <file src="toCopy/start.bat" targetdir="$INSTALL_PATH"><os family="windows"/></file>
      <file src="toCopy/logo_32x32.ico" targetdir="$INSTALL_PATH"><os family="windows"/></file>
      <file src="toCopy/un_32x32.ico" targetdir="$INSTALL_PATH"><os family="windows"/></file>
      <file src="toCopy/logo_32x32.png" targetdir="$INSTALL_PATH"><os family="unix"/></file>
      <file src="toCopy/un_32x32.png" targetdir="$INSTALL_PATH"><os family="unix"/></file>
    </pack>
    <pack name="Data" required="yes" preselected="yes" hidden="no">
      <description>The necessary data</description>
      <file src="toCopy/bcon-fine" targetdir="$INSTALL_PATH" />
      <file src="toCopy/CHAMPIGNON.bcn" targetdir="$INSTALL_PATH" />
      <file src="toCopy/middleLeg.vtk" targetdir="$INSTALL_PATH" />
      <file src="toCopy/MIR-bottom_master_0.vtk.tmp.obj" targetdir="$INSTALL_PATH" />
      <file src="toCopy/MIR-top_master_0.vtk.tmp.obj" targetdir="$INSTALL_PATH" />
      <file src="toCopy/PacMan.stl" targetdir="$INSTALL_PATH" />
      <file src="toCopy/Shroom.stl" targetdir="$INSTALL_PATH" />
      <file src="toCopy/test.stl" targetdir="$INSTALL_PATH" />
      <file src="toCopy/weldedSpheres.vtk" targetdir="$INSTALL_PATH" />
    </pack>
  </packs>
  <!-- Launch the executable after installation -->
</installation>

```

