

**CDPP - 3DVIEW**

Réf. :CNES/CDPP-3DView/PRD/DOC/CG

**CDPP-SR-32600-533-GFI**

Edition : 01 Date : 21/07/2016

Révision : 00 Date : 21/07/2016

MT : X Code diffusion : E

**DOCUMENT DE CONCEPTION  
3DVIEW**

<b>Rédigé par :</b> BEIGBEDER Laurent CAUSSARIEU Stéphane	GFI INFORMATIQUE GFI INFORMATIQUE	le :	
<b>Validé par :</b> TONIUTTI Jean-Philippe	GFI INFORMATIQUE	le :	

## BORDEREAU D'INDEXATION

CONFIDENTIALITE :  
**NC**

MOTS CLES : **Conception, 3DView, CDPP**

TITRE DU DOCUMENT :

**Document de conception  
3DView**

AUTEUR(S) :

**BEIGBEDER Laurent**

**GFI INFORMATIQUE**

**CAUSSARIEU Stéphane**

**GFI INFORMATIQUE**

RESUME : **Document de conception du logiciel CDPP 3DView**

DOCUMENTS RATTACHES : **Ce document vit seul.**

LOCALISATION :

**CNES/CDPP-3DView/PRD/DOC**

VOLUME : **1**

NBRE TOTAL DE PAGES : **92**

DOCUMENT COMPOSITE : **N**

LANGUE : **FR**

DONT PAGES LIMINAIRES : **10**

NBRE DE PAGES SUPPL. : **0**

GESTION DE CONF. : **NG**

RESP. GEST. CONF. :

CAUSE D'EVOLUTION : **Prise en compte des retours recette de la version logicielle V1.11**

CONTRAT : **Bon de commande MARCHÉ SOUS ACCORD-CADRE N° 151283**

SYSTÈME HÔTE :

**Microsoft Word 11.0 (11.0.5604)**

**L:\CLASSE1\Modèles word\GDOC V3.1.8\ModeleGDOCIndus\_2015.dot**

**Version GDOC : v3.1.8**

## DIFFUSION EXTERNE

Nom	Sigle	Bpi	Observations
DUFOURG Nicolas	DCT/ME/EU	612	
DURAND Joëlle	DCT/ME/EU	612	

## DIFFUSION INTERNE

Nom	Sigle	Observations
BEIGBEDER Laurent	GFI INFORMATIQUE	
CAUSSARIEU Stéphane	GFI INFORMATIQUE	
POPESCU Daniel	GFI INFORMATIQUE	
TONIUTTI Jean-Philippe	GFI INFORMATIQUE	

## MODIFICATION

Ed.	Rév.	Date	Référence, Auteur(s), Causes d'évolution
01	00	21/07/2016	CNES/CDPP-3DView/PRD/DOC/CG BEIGBEDER Laurent GFI INFORMATIQUE CAUSSARIEU Stéphane GFI INFORMATIQUE Prise en compte des retours recette de la version logicielle V1.11
00	01	22/06/2016	CNES/CDPP-3DView/PRD/DOC/CG BEIGBEDER Laurent GFI INFORMATIQUE CAUSSARIEU Stéphane GFI INFORMATIQUE Mise à jour au titre de la version logicielle V1.11
00	00	04/12/2015	CNES/CDPP-3DView/PRD/DOC/CG BEIGBEDER Laurent GFI INFORMATIQUE Création du document

## SOMMAIRE

<b>GLOSSAIRE ET LISTE DES PARAMETRES AC &amp; AD .....</b>	<b>1</b>
<b>1. GENERALITES .....</b>	<b>2</b>
<b>1.1. DOCUMENTS APPLICABLES .....</b>	<b>2</b>
<b>1.2. DOCUMENTS DE REFERENCE .....</b>	<b>2</b>
<b>2. BUT DU DOCUMENT .....</b>	<b>3</b>
<b>3. INTRODUCTION.....</b>	<b>4</b>
<b>4. ARCHITECTURE.....</b>	<b>5</b>
<b>4.1. SCHEMA .....</b>	<b>5</b>
<b>4.1.1. 3DView CDPP et accès aux services externes .....</b>	<b>5</b>
<b>4.1.2. Architecture Interne .....</b>	<b>6</b>
<b>4.1.3. Matrice de flux.....</b>	<b>7</b>
<b>4.1.4. Architecture matérielle.....</b>	<b>7</b>
<b>4.2. DESCRIPTION DE L'ARCHITECTURE.....</b>	<b>7</b>
<b>4.2.1. La partie client .....</b>	<b>8</b>
<b>4.2.2. Les web services.....</b>	<b>8</b>
4.2.2.1. Le service d'orbitographie .....	8
4.2.2.2. Le service de changement de repère .....	8
4.2.2.3. Le service de modèles.....	8
4.2.2.4. Le service d'analyse des fichiers résultats .....	9
4.2.2.4.1. Stockage des Cubes sur le serveur .....	9
4.2.2.4.2. Interpolation des simulations 3D statiques le long d'une courbe .....	9
4.2.2.4.3. Génération de coupe à partir des simulations 3D statiques .....	9
4.2.2.5. Technologies utilisées .....	9
<b>4.2.3. La mise à jour des données .....</b>	<b>10</b>
<b>4.3. ARCHITECTURE LOGICIELLE .....</b>	<b>10</b>
<b>4.4. ORGANISATION DES REPERTOIRES .....</b>	<b>10</b>
<b>4.4.1. Arborescence DPC .....</b>	<b>11</b>
<b>4.4.2. Arborescence Web .....</b>	<b>13</b>
<b>5. DESCRIPTION DE LA GESTION DES FICHIERS D'ORBITE ET D'ATTITUDE .....</b>	<b>14</b>
<b>5.1. LES FICHIERS UTILISES POUR LES MISSIONS .....</b>	<b>14</b>
<b>5.1.1. Les données de type ESA .....</b>	<b>14</b>
5.1.1.1. Rosetta .....	15
5.1.1.2. Mars-Express .....	15
5.1.1.3. Venus-Express .....	16
<b>5.1.2. Les données venant du SSCWeb .....</b>	<b>16</b>
<b>5.1.3. Les autres types de donnée.....</b>	<b>17</b>
5.1.3.1. Cassini.....	17
5.1.3.1.1. Les fichiers d'orbite .....	17

5.1.3.1.2. Les fichiers d'attitude .....	17
5.1.3.2. Galileo .....	17
5.1.3.2.1. Les fichiers d'orbite .....	17
5.1.3.3. Ulysses .....	18
5.1.3.3.1. Les fichiers d'orbite .....	18
5.1.3.4. Stereo .....	18
5.1.3.4.1. Les fichiers d'orbite .....	18
5.1.3.5. SOHO .....	19
5.1.3.5.1. Les fichiers d'orbite .....	19
5.1.3.6. Geotail .....	20
5.1.3.6.1. Les fichiers d'orbite .....	20
5.1.3.7. Messenger .....	20
5.1.3.7.1. Les fichiers d'orbite .....	20
5.1.3.8. INTERBALL .....	21
5.1.3.8.1. Les fichiers d'orbite .....	21
5.1.3.8.2. Les fichiers d'attitude .....	21
5.1.3.9. GIOTTO .....	22
5.1.3.9.1. Les fichiers d'orbite .....	22
5.1.3.9.2. Les fichiers d'attitude .....	22
5.1.3.10. JASON2 .....	23
5.1.3.10.1. Les fichiers d'orbite .....	23
5.1.3.10.2. Les fichiers d'attitude .....	24
5.1.3.10.3. Les fichiers d'orientation des générateurs solaires .....	25
5.1.3.11. PICARD .....	26
5.1.3.11.1. Les fichiers d'orbite .....	26
5.1.3.11.2. Les fichiers d'attitude .....	27
5.1.3.12. MRO .....	28
5.1.3.12.1. Les fichiers d'orbite .....	28
5.1.3.12.2. Les fichiers d'attitude .....	28
5.1.3.13. Juno .....	28
5.1.3.13.1. Les fichiers d'orbite .....	28
5.1.3.14. JUICE .....	28
5.1.3.14.1. Les fichiers d'orbite .....	28
5.1.3.15. HELIOS .....	29
5.1.3.15.1. Les fichiers d'orbite .....	29
5.1.3.16. Pionner .....	29
5.1.3.16.1. Les fichiers d'orbite .....	29
5.1.3.17. Phobos2 .....	29
5.1.3.17.1. Les fichiers d'orbite .....	29
5.1.3.18. Chandrayan .....	29
5.1.3.18.1. Les fichiers d'orbite .....	29
5.1.3.19. DSCOVR .....	30
5.1.3.19.1. Les fichiers d'orbite .....	30
5.1.3.20. GAIA .....	31
5.1.3.20.1. Les fichiers d'orbite .....	31
5.1.3.21. Herschel .....	31
5.1.3.21.1. Les fichiers d'orbite .....	31
5.1.3.22. Planck .....	32
5.1.3.22.1. Les fichiers d'orbite .....	32
5.1.3.23. Xmm .....	33
5.1.3.23.1. Les fichiers d'orbite .....	33
<b>5.2. LES SERVEURS .....</b>	<b>34</b>
<b>5.2.1. Le serveur NAIF .....</b>	<b>34</b>

<b>5.2.2. Le rapatriement des fichiers.....</b>	<b>34</b>
<b>5.2.3. Les fichiers scripts .....</b>	<b>34</b>
<b>5.2.4. Les fichiers de configuration .....</b>	<b>35</b>
<b>5.2.5. Les missions mises à jour .....</b>	<b>35</b>
5.2.5.1. Cassini.....	35
5.2.5.1.1. Fichiers d'orbite.....	35
5.2.5.2. Rosetta .....	36
5.2.5.2.1. Les fichiers d'orbite .....	36
5.2.5.2.2. Les fichiers d'attitude .....	36
5.2.5.2.3. Les fichiers d'orientation des panneaux solaires .....	36
5.2.5.3. Venus-Express .....	36
5.2.5.3.1. Les fichiers d'orbite .....	36
5.2.5.3.2. Les fichiers d'attitude .....	37
5.2.5.3.3. Les fichiers d'orientation des panneaux solaires .....	37
5.2.5.4. Mars-Express .....	37
5.2.5.4.1. Les fichiers d'orbite .....	37
5.2.5.4.2. Les fichiers d'attitude .....	37
5.2.5.4.3. Les fichiers d'orientation des panneaux solaires .....	38
5.2.5.5. MRO .....	38
<b>5.2.6. Fichiers d'index .....</b>	<b>38</b>
5.2.6.1. naiffiles.lst.....	38
5.2.6.2. sccoverage.lst.....	39
<b>6. SELECTION ET AFFICHAGE DES DONNEES .....</b>	<b>40</b>
<b>6.1. GENERALITES .....</b>	<b>40</b>
<b>6.2. LE CLIENT 3D .....</b>	<b>40</b>
<b>6.2.1. Lancement.....</b>	<b>40</b>
<b>6.2.2. Architecture de l'application Java.....</b>	<b>43</b>
6.2.2.1. Liste des principaux composants utilisés .....	43
6.2.2.2. Les composants .....	43
6.2.2.2.1. Dépendances.....	43
6.2.2.2.2. L'API Java2 Standard Edition.....	44
6.2.2.2.3. L'API java3D .....	44
6.2.2.2.4. L'API Java Media Framework .....	45
6.2.2.2.5. L'API JFreeChart .....	45
6.2.2.2.6. L'API Vrml97 .....	45
6.2.2.2.7. L'API 3ds .....	45
6.2.2.2.8. L'API netCDF-JAVA.....	45
6.2.2.2.9. JSAMP.....	45
6.2.2.2.10. SGP4 .....	46
6.2.2.2.11. CDAS Web Services Developer's Kit. ....	46
6.2.2.2.12. OkHttp .....	46
6.2.2.3. L'arborescence du package mm3dview .....	46
<b>6.2.3. Implémentation des fonctionnalités .....</b>	<b>50</b>
6.2.3.1. Composition de la scène 3D.....	50
6.2.3.2. Eclairage du soleil .....	51
6.2.3.3. Orientation des instruments.....	51
6.2.3.4. Sauvegarde de scénario.....	52
6.2.3.5. Export des données.....	52
6.2.3.6. Génération d'images .....	53
6.2.3.7. Génération de films .....	53
6.2.3.8. Affichages 2D .....	53

6.2.3.9. Affichage des modèles de analytique intégrés au client 3D .....	54
6.2.3.9.1. Modèles de chocs/magnetopause/MPB .....	54
6.2.3.9.1.1. Modèles de Mercure .....	54
6.2.3.9.1.1.1. Choc .....	54
6.2.3.9.1.1.2. Magnétopause .....	55
6.2.3.9.1.2. Modèle de Venus .....	55
6.2.3.9.1.2.1. Choc .....	55
6.2.3.9.1.3. Modèles de la terre .....	55
6.2.3.9.1.3.1. Onde de choc .....	55
6.2.3.9.1.3.2. Magnétopause de Shue .....	55
6.2.3.9.1.4. Modèles de Mars .....	56
6.2.3.9.1.4.1. Choc .....	56
6.2.3.9.1.4.2. Magnetic pile-up boundary .....	56
6.2.3.9.1.5. Modèles de Saturne .....	56
6.2.3.9.1.5.1. Choc .....	56
6.2.3.9.1.5.2. Magnétopause .....	56
6.2.3.9.2. Autres modèles physiques .....	57
6.2.3.9.2.1. Modèles du soleil .....	57
6.2.3.9.2.1.1. Spirale de parker .....	57
6.2.3.9.2.1.2. La couche de courant .....	57
6.2.3.9.2.1.3. Les fronts de CMEs .....	57
6.2.3.9.2.2. Modèles de la terre .....	57
6.2.3.9.2.3. Format des coquilles des ceintures .....	57
6.2.3.9.2.4. Format de la SAA .....	58
6.2.3.10. Etoiles .....	58
6.2.3.11. Génération de timetables .....	59
6.2.3.12. Ajout d'objets inconnus de 3DView à partir d'un fichier utilisateur ..	59
6.2.3.13. Ajout de l'attitude d'un objet de 3DView à partir d'un fichier	
utilisateur .....	59
6.2.3.14. Ajout d'une ligne de champ à partir d'un fichier utilisateur .....	59
6.2.3.15. Ajout de champs de vue d'un instrument à partir d'un fichier	
utilisateur .....	59
6.2.3.16. Décalage des objets dans le temps .....	60
6.2.3.17. Ajout de satellites de la terre à partir d'un fichier utilisateur au	
format TLE .....	60
<b>6.3. LES WEB SERVICES 3DVIEW .....</b>	<b>61</b>
<b>6.3.1. Description .....</b>	<b>63</b>
<b>6.3.2. Implémentation client .....</b>	<b>63</b>
<b>6.3.3. Implémentation serveur .....</b>	<b>63</b>
6.3.3.1. Fichiers de l'application web .....	64
6.3.3.1.1. bodies.properties .....	64
6.3.3.1.2. services.properties .....	66
6.3.3.1.3. conf.properties .....	66
6.3.3.2. Configuration des données .....	67
6.3.3.2.1. groundstations.conf .....	67
6.3.3.2.2. scparams.conf .....	67
6.3.3.2.3. starfiles.conf .....	67
6.3.3.2.4. scinstruments.conf .....	68
6.3.3.2.5. groundlabels.conf .....	68
6.3.3.2.6. frames.conf .....	69
6.3.3.2.7. crmaps.conf .....	69
6.3.3.2.8. projmaps.conf .....	70

<b>6.4. AFFICHAGE DES DONNEES AU FORMAT IMPEX.....</b>	<b>71</b>
<b>6.4.1. Les types de données .....</b>	<b>73</b>
6.4.1.1. Les données d'observation.....	73
6.4.1.1.1. AMDA, CLWeb .....	73
6.4.1.1.2. CDAWeb.....	73
6.4.1.1.3. Cluster Science Archive .....	73
6.4.1.1.4. Par fichier .....	73
6.4.1.2. Les données de simulation.....	74
6.4.1.2.1. Les données precomputed.....	74
6.4.1.2.2. Les données on the fly.....	74
<b>6.4.2. L'arbre de sélection .....</b>	<b>74</b>
6.4.2.1. Les nœuds génériques .....	74
6.4.2.2. La construction de l'arbre .....	75
6.4.2.2.1. Les nœuds venant des tree.xml .....	75
6.4.2.2.2. Les nœuds venant du CDAWeb .....	75
6.4.2.2.3. Les nœuds venant du CSA .....	75
6.4.2.2.4. Les nœuds venant d'Amda et CLWeb .....	75
6.4.2.2.5. Les nœuds venant des web services LATMOS, FMI, SINP et 3DView ..	76
6.4.2.3. Les filtres .....	76
6.4.2.4. Les interfaces d'entrée des Web services.....	76
6.4.2.4.1. JWSInputGetPanel (SINP, FMI, LATMOS, 3DView) .....	77
6.4.2.4.2. JWSInputCalcPanel (SINP, 3DView) .....	78
6.4.2.4.3. JWSInputFieldLinePanel (3DView, FMI, LATMOS, SINP) .....	79
6.4.2.4.4. getTimeTableList et getTimeTable (AMDA, CLWeb) .....	79
6.4.2.4.5. getParameterList et getParameter (AMDA, CLWeb).....	79
<b>6.4.3. Lecture et affichage des données .....</b>	<b>79</b>
6.4.3.1. Les 3Dcubes .....	80
6.4.3.2. Les 2Dcuts .....	80
6.4.3.3. Les TimeSeries .....	80
6.4.3.4. Les Spectrogrammes .....	81
6.4.3.5. Les timetables .....	81
6.4.3.6. Les lignes de champs.....	81
<b>6.4.4. Communication spécifique avec AMDA .....</b>	<b>82</b>
6.4.4.1. Afficher des données d'AMDA dans 3DView.....	82
<b>6.4.5. Les paramètres du client.....</b>	<b>82</b>
<b>6.4.6. Gestion du cache/espace commun .....</b>	<b>83</b>
6.4.6.1. L'espace commun .....	83
6.4.6.2. Le cache.....	83

## GLOSSAIRE ET LISTE DES PARAMETRES AC & AD

AMDA	<i>Automated Multi Dataset Analysis</i>
BT	Bilan Technique
CDPP	Centre de Données de Physique des Plasmas
CLWEB	Web version of CL
CNES	Centre National d'Etudes Spatiales
COROT	Convection, Rotation et Transits planétaires
DPC	Data Processing Center
ESA	European Space Agency
ESOC	European Space Operations Centre
FMI	Finnish Meteorological Institute
IMPEX	Integrated Medium for Planetary Exploration
JAX-RPC	Java API for XML Remote Procedure Call
JAX-WS	Java API for XML Web Services or JAX-RPC 2.0
JDK	Java Development Kit
JMF	Java Media Framework
JRE	Java Runtime Environment
LATMOS	Laboratoire Atmosphères, Milieux, Observations Spatiales
MEX	Mars-Express
NAIF	Navigation and Ancillary Information Facility
NASA	National Aeronautics and Space Administration
NetCDF	Network Common Data Form
ROS	Rosetta
SINP	Skobeltsyn Institute of Nuclear Physics
SOAP	Simple Object Access Protocol
SPICE	Spacecraft ephemeris, Planet location, Instrument, Pointing and Events kernels
URL	Uniform Resource Locator
VEX	Venus-Express
VRML	Virtual reality Modeling Language
WAR	Web Application ARchive
WSDL	Web Services Description Language

**Liste des paramètres AC :**

**Liste des paramètres AD :**

## 1.GENERALITES

---

### 1.1.DOCUMENTS APPLICABLES

DA1 Cf. les DA du Répertoire de la documentation 3DView/CDPP  
L. BEIGBEDER, 21/07/2016, Issue 01, Rev. 00  
**CDPP-LI-32600-532-GFI**

### 1.2.DOCUMENTS DE REFERENCE

DR1 Cf. les DR du Répertoire de la documentation 3DView/CDPP  
L. BEIGBEDER, 21/07/2016, Issue 01, Rev. 00  
**CDPP-LI-32600-532-GFI**

## 2.BUT DU DOCUMENT

---

Ce document a pour but de présenter la conception du logiciel CDPP 3Dview.

Le chapitre 3 (Introduction) rappelle le contexte à l'origine de la création du logiciel et ses grandes fonctions.

Le chapitre 4 présente l'architecture matérielle et logicielle.

Le chapitre 5 décrit la gestion des fichiers d'orbite et d'attitude.

Le chapitre 6 décrit l'interface web services et l'affichage 3D.

## 3.INTRODUCTION

---

CDPP 3Dview est une application interactive de visualisation 3D animée de trajectoires et d'attitudes de sondes interplanétaires du système solaire ainsi que de données physiques.

L'outil, basé sur une architecture SOA, est orienté science avec l'affichage de modèles scientifiques et l'intégration de web services.

Il est issu de la version 3DView IMPEX. Elle était destinée à comparer des modèles théoriques issus de simulation de différents laboratoires (LATMOS, FMI, SINP) à des données d'observation d'AMDA et du CDAWeb.

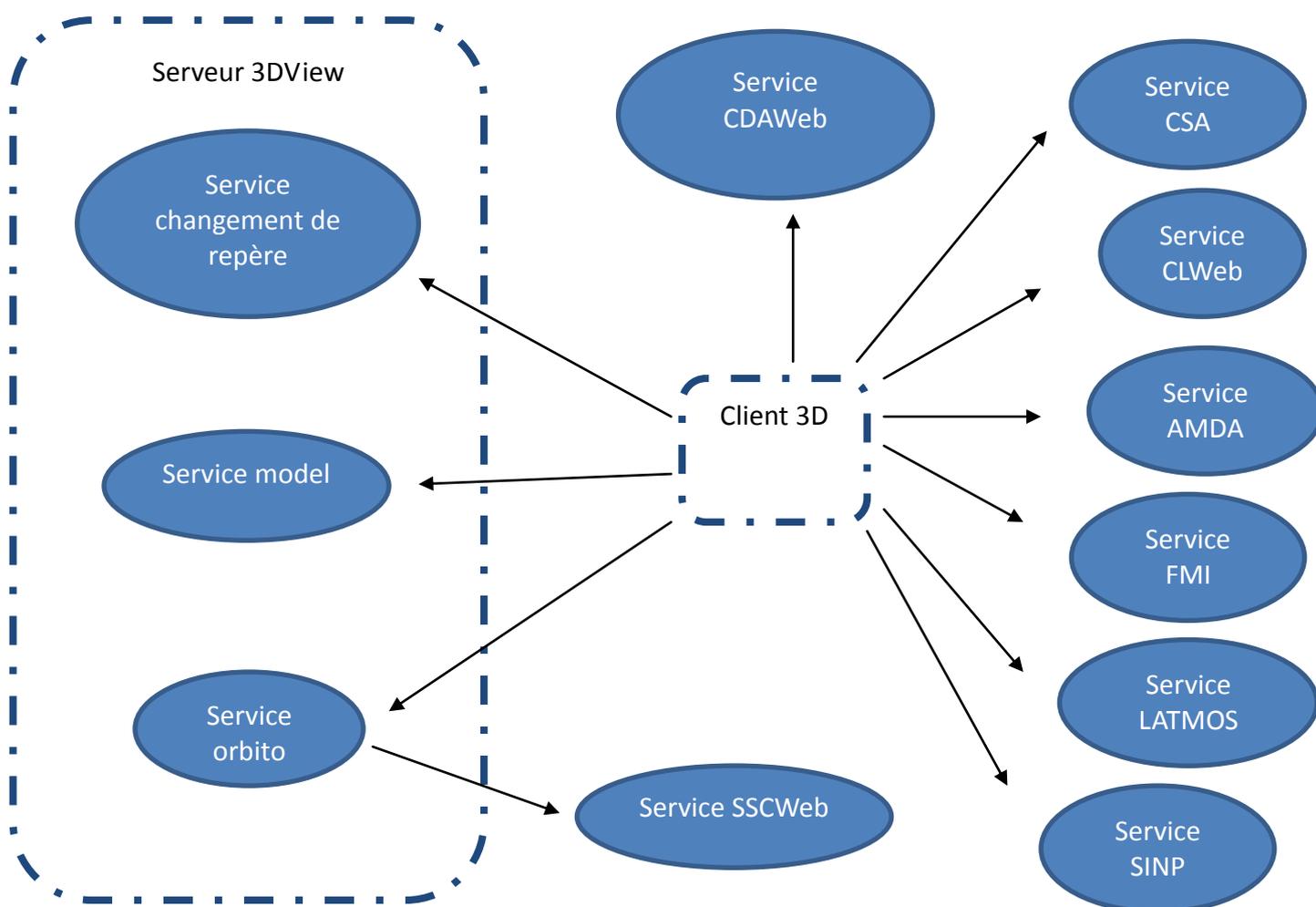
Cette version, en ajoutant de nouvelles fonctionnalités, ouvre le 3DView à des secteurs autres que le plasma, comme la planétologie.

CDPP 3Dview est un outil « léger », interactif et intuitif dont la prise en main est facile et permet une grande autonomie d'utilisation.

## 4.ARCHITECTURE

### 4.1.SHEMA

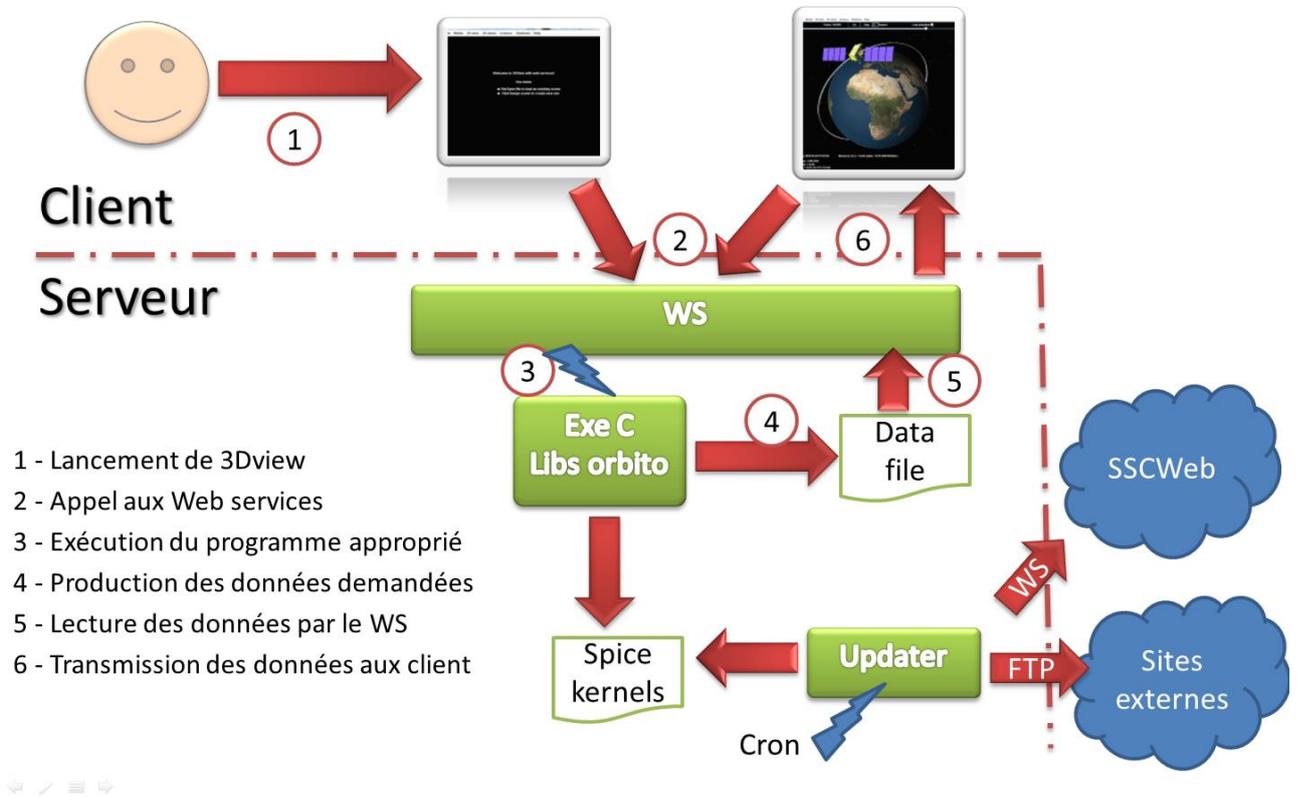
#### 4.1.1. 3DView CDPP et accès aux services externes



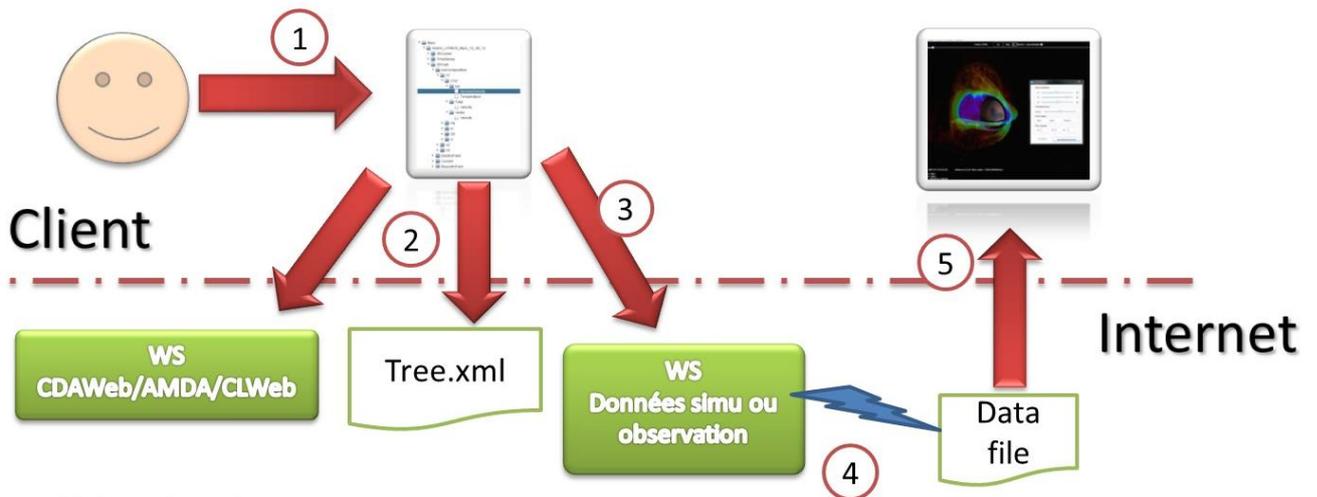
Ce schéma résume les différents acteurs externes en communication avec 3DView. Les connexions se font majoritairement depuis le client 3D donc depuis la machine de l'utilisateur.

### 4.1.2. Architecture Interne

Affichage de la scène



Ajout de données science



- 1 - Affichage de l'arbre
- 2 - Construction de l'arbre à partir de tree.xml ou des web services
- 3 - Demande d'une donnée après sélection dans l'arbre
- 4 - Sélection ou production des données demandées
- 5 - Lecture des données du fichier et affichage dans la scène

### 4.1.3. Matrice de flux

Entité initiatrice	Flux	Entité réceptrice	Port	Protocole	Description
Client 3D	=>	Server 3DView	80	HTTP	Web service pour obtenir des données d'orbite et attitude, champs magnétique, données statiques (modèles 3D, ...)
Serveur 3DView	=>	Fournisseur de données	21	FTP	Fichier d'orbite, d'attitude à mettre à jour
Serveur 3DView	=>	Sscweb	80	HTTP	Web service pour obtenir des données d'orbite
Client 3D	=>	AMDA, CLWeb, CDAWeb, FMI, LATMOS, SINP, CSA	80	HTTP	Web service pour obtenir des données d'observation et simulation, accès direct aux fichiers de données
Tous	=>	Serveur 3DView	80	HTTP	Web service d'orbite et d'attitude, changement de repère, champ magnétique

### 4.1.4. Architecture matérielle

L'ensemble est contenu dans une machine Virtuelle au format VirtualBox avec comme Système d'exploitation hôte CentOS 6.4 64bits. Un disque de 50Go, une mémoire de 2Go et un accès réseau par pont sont configurés. Elle peut être hébergée sur un ordinateur à architecture x86 supportant le 64 bits avec une technologie de virtualisation.

La machine virtuelle héberge les logiciels openJdk 7 et tomcat7 sur le port 8080. Une règle de firewall redirige le flux sur le port 80.

## 4.2. DESCRIPTION DE L'ARCHITECTURE

Le schéma précédent présente le système de fonctionnement du logiciel 3Dview et des web services.

Le logiciel est constitué de 3 composants principaux :

- Le client 3D lancé par l'utilisateur,
- Un serveur de web service d'orbitographie, de changement de repère et de modèles physiques,
- De procédures de mise à jour des données.

### 4.2.1. La partie client

Le client 3D est lancé par l'utilisateur via une page web ou directement depuis l'icône du bureau, disponible après le premier lancement. Il se connecte aux web services pour obtenir les données à tracer.

Les langages et technologies utilisés sont :

- JAVA 7
- Swing, JAX-WS, Java3D, JMF, netCDF, XML

Le client est testé pour fonctionner Windows, MacOSX et Linux, en 32 et 64 bits.

Pour cela, la version JDK1.6 est utilisée ainsi qu'un lancement spécifique sur MacOSX pour désactiver le java3D présent par défaut.

### 4.2.2. Les web services

Les web service 3DView fournissent des données au client mais sont aussi disponible sur internet à toute autre application voulant y faire appel.

#### 4.2.2.1. Le service d'orbitographie

Toute demande au service génère une consultation des données du serveur stockées sous deux formes principales : des fichiers de configuration et des fichiers de données. Pour les fichiers de configuration, un appel JAVA direct est effectué. Pour les fichiers de données, un appel à un exécutable natif (spécifique à la plateforme) génère un fichier qui est ensuite interprété.

#### 4.2.2.2. Le service de changement de repère

Il permet de convertir un ensemble de point d'un repère donné dans un autre système de coordonnées.

Lors de la consultation du service, la couche J2EE fait appel à un exécutable C se basant sur des SPICE « frame kernels » au format texte.

Ces repères sont définis dans le fichier \$HOME/i3dv\_files/orbitfiles/spice/impex.tf et listés dans le fichier \$HOME/i3dv/server/web/conf/frames.conf

#### 4.2.2.3. Le service de modèles

Le service de modèle permet d'obtenir des lignes de champs magnétiques en faisant appel à des exécutables C correspondant à chaque modèle.

#### 4.2.2.4. Le service d'analyse des fichiers résultats

Ce service permet de traiter des cubes au format LATMOS. Le client commande au serveur de télécharger le cube sur son espace de stockage puis lui demande de réaliser un traitement sur cette ressource.

##### 4.2.2.4.1. Stockage des Cubes sur le serveur

Les fichiers sont stockés sur le serveur dans le répertoire \$HOME/i3dv\_files/commonres.

Dans ce répertoire, il y aura des cubes de l'ordre du giga. Ils sont classés dans une arborescence correspondant à leur ressource SPASE (ex. IMPEX/NumericalOutput/ LATMOS/Hybrid/Mars\_13\_02\_13/CO2/3D).

Si le fichier existe déjà, il n'est pas re-téléchargé mais il faut contrôler le volume disponible sur le disque et peut être faire un montage NFS pour ce répertoire si l'espace disque vient à manquer. Le chemin de ce répertoire est modifiable dans un fichier conf dans le fichier WAR (WEB-INF/classes/conf.properties).

##### 4.2.2.4.2. Interpolation des simulations 3D statiques le long d'une courbe

Ce service permet l'interpolation le long d'une courbe dans un cube en faisant appel à un algorithme d'interpolation trilinéaire.

##### 4.2.2.4.3. Génération de coupe à partir des simulations 3D statiques

Ce service permet de générer une grille formant une surface à l'intérieur du cube. L'algorithme d'interpolation trilinéaire est ensuite utilisé pour chaque point de la grille.

Le long d'une courbe dans un cube en faisant appel à un algorithme d'interpolation trilinéaire.

#### 4.2.2.5. Technologies utilisées

Les langages et technologies utilisés sont :

- JAVA 6, C, FORTRAN,
- J2EE, JAX-WS, SPICELIB et OASWLIB.

### 4.2.3. La mise à jour des données

C'est un programme java qui est lancé tous les jours automatiquement. Il se connecte aux différents serveurs de données et pour vérifier la présence de nouvelles données et les rapatrier si nécessaire.

Les nouvelles données sont soit intégrées directement soit converties au format SPICE.

Les langages et technologies utilisés sont :

- JAVA 6, Shell
- FTP, SPICELIB, Apache commons.

## 4.3. ARCHITECTURE LOGICIELLE

Pour fonctionner, le serveur 3Dview CDPP nécessite :

- Le système d'exploitation Linux CentOS 6.3.
- Une arborescence contenant les fichiers de "base", décrite ci-dessous,
- Le serveur web tomcat 7,
- Le Java Development Kit 1.6 ou supérieur.

Le client 3DView lui, requiert :

- Un navigateur,
- Le Java Runtime Environment. 1.7\_045 ou supérieur,
- Des drivers graphiques openGL1.3+ ou DirectX9.

## 4.4. ORGANISATION DES REPERTOIRES

L'arborescence serveur est divisée en deux parties :

- Une partie "DPC" (data processing center) contenant les fichiers de données (configuration et noyaux SPICE) et les procédures de mise à jour,
- Une application web dans tomcat.

### 4.4.1.Arborescence DPC

Dans le répertoire home de l'utilisateur, elle stocke les procédures de mise à jour et les fichiers de données :

```

i3dv
|__client
|__install
|__server
| |__dpc
| | |__bin
| | | |__linux
| | | |__win
| | | |__solaris
| | |__lib
| | |__scripts
| | |__dist
| |__web
| | |__conf
| | |__include
| | |__java
| | |__native
| | |__static
i3dv_files
|__carringtonmap
|__commonres
|__fitcatalog
|__tmp
|__logs
|__orbitfiles
| |__spice
| | |__art
| | | |__mro
| | | |__ros
| | | |__mex
| | | |__vex
| | | |__corot
| | | |__work
| | | |__jason2
| | | |__picard
| | | |__cass
|__conf
| |__spice

```

Nous précisons le rôle de chaque répertoire :

- **I3dv** : répertoire contenant les sources de tout l'applicatif ainsi que les objets compilés
  - Client : contient les sources java du client3D
  - Install : contient les scripts de configuration de l'environnement et d'installation
  - Server : contient tous les sources du serveur (DPC, services, web app)
    - **dpc** : répertoire contenant les outils de mise à jour des données. Il est subdivisé en 4 sous-répertoires :
      - **bin** : exécutable spice pour la création et manipulation des fichiers kernels spice,
      - **scripts** : contient les scripts de mise à jour des données,
      - **dist** : application JAVA de mise à jour des données,
      - **lib** : bibliothèques JAVA utilisées par l'application de mise à jour.
    - **Web** : Contient l'appliweb et les exécutable spice natifs en C/Fortran
      - **Include** : headers généraux C
      - **Java** : sources java de l'appli web
      - **Native** : sources C/fortran des exécutable spice
      - **Static** : pages statiques du site web (HTML, jsp, ...)
- **I3dv\_files** : contient les logs et les données
  - **conf** : les fichiers de configuration des mises à jour sont mis ici.
  - **logs** : répertoire de résidence des fichiers log de mise à jour et conversion des fichiers.
  - **carringtonmap** : lien vers les cartes carrington
  - **commonres** : stocke les cube lors des traitements locaux
  - **fitcatalog** : lien vers les catalogues CME
  - **orbitfiles** : répertoire de résidence des fichiers d'orbite et d'attitude. A la racine se trouve le fichier qui indexe les fichiers d'orbite et d'attitude : naiffiles.lst.  
Il contient le sous-répertoire *spice* pour les fichiers de type naif. Ce répertoire contient un fichier kernels.txt contenant tous les kernels à charger au démarrage de l'application. Le dossier contient aussi les noyaux des planètes (de421.bsp) et des satellites (\*.bsp). Le dossier art contient les corps artificiels tels que MEX, Rosetta, Ulysse, Stereo, Cassini, ...
  - **tmp** : pour mettre les fichiers temporaires. Non utilisé pour l'instant.

## 4.4.2.Arborescence Web

L'arborescence web est contenue dans un WAR et installé dans tomcat.

```
|__client3d  
| |__...  
|__js  
|__models  
|__pages  
|__css  
|__WEB-INF  
| |__classes  
| | |__...  
|__images  
|__other  
|__samples
```

Nous précisons le rôle de chaque répertoire :

- **client3d** : répertoire contenant toutes les ressources nécessaires à l'exécution du client3D.
- **models** : contient les modèles 3D des comètes et des satellites artificiels. Le nom des modèles comète sont nommés <NAIF\_ID>\_model.3ds. Les nom des satellites peuvent être déclinés en deux parties : le corps et les panneaux solaires : sc<NAIF\_ID>\_(body|sa).3ds. Si les modèles de satellite 3ds ont des textures, elles doivent se situer dans le répertoire models/texture/<nom du fichier 3ds>/\*. \*
- **Js, css, pages, images, other, samples** : fichiers web statiques du site : images, pages web html, CSS, ...
- **WEB-INF/classes** : Contient les ressources java des web services et les fichiers de configuration

## 5. DESCRIPTION DE LA GESTION DES FICHIERS D'ORBITE ET D'ATTITUDE

Pour pouvoir afficher des trajectoires, des attitudes de sondes comme Rosetta ou Cassini, il faut connaître leur position. Pour obtenir ces données, des fichiers préparés par la NASA ou l'ESA sont utilisés. Une connexion directe au serveur SSCWeb permet aussi de créer ou de mettre à jour des fichiers d'orbite de certains satellites terrestres.

### 5.1. LES FICHIERS UTILISES POUR LES MISSIONS

Pour toutes les données les noyaux SPICE (spice kernels) sont utilisés. Ces noyaux peuvent être générés ou bien téléchargés directement sur le site de la NAIF.

Pour chaque mission, les fichiers sont de types différents. Néanmoins, nous pouvons classer ces fichiers en trois catégories. Les fichiers normés ESA qui ont une nomenclature spécifique, les fichiers importés depuis le SSCWeb, et les autres.

#### 5.1.1. Les données de type ESA

Ces fichiers sont normés de la manière suivante :

**ffff\_sssddd\_Dwxyymmddhmmss\_vvvv**

Voici la description tirée du document des conventions de nommage des fichiers "ancillary" de l'ESOC :

**ffff** File type identifier

ORHM = orbit file heliocentric Mars Express

ORMM = orbit file marscentric Mars Express

ORMF = orbit file marscentric frozen orbit

ORHR = orbit file heliocentric Rosetta

ATNM = attitude file nominal Mars Express

ATNR = attitude file nominal Rosetta

EVTM = event file Mars Express

EVTF = event file frozen orbit (Mars Express)

VILM = visibility file Lander Mars Express

OASW = orbit and attitude data access software

**sss** Data source identifier

FDx = ESOC Flight Dynamics

PST = ESTEC Project Science Team Mars Express (for SPICE files)

**ddd** Data destination identifier

MMx = Mars Express Mission System (DDS)

PIX = PI Teams (for SPICE files from PST)

**D** Data file

**w** format identifier

B = binary data

A = ASCII data

**x** Data type identifier

\_ = (underscore) orbit data

P = predicted attitude data

R = reconstructed attitude data

**yymmddhhmmss** Start time of data in file

Except for orbit files where the time stamp is replaced by  
Twelve (12) underscores

**vvvvv** Version number

### 5.1.1.1.Rosetta

Les fichiers récupérés pour cette mission sont les fichiers :

- ATNR : attitude nominale
- ATPR : attitude prédite
- ORHR : orbite centrée sur le soleil
- ORER : orbite centrée sur la terre au moment du premier survol de cette dernière
- ORFR : idem, 2<sup>ème</sup> survol
- ORGR : idem 3<sup>ème</sup> survol
- ORMR : orbite centrée sur mars au moment de son survol
- RORB : orbite de rosetta reconstruite centrée sur le soleil à partir de 2014
- RATT : attitude de rosetta reconstruite centrée sur le soleil à partir de 2014

### 5.1.1.2.Mars-Express

Les fichiers récupérés pour cette mission sont les fichiers :

- ATNM : Attitude nominale
- ATPM : Attitude prédite
- ORHM : orbite centrée sur le soleil (phase croisière)
- ORMM : orbite centrée sur mars (phase orbite)

Ces fichiers sont souvent mis à jour sur le serveur. Un ORMM couvre au maximum la durée d'un mois.

### 5.1.1.3.Venus-Express

Les fichiers récupérés pour cette mission sont les fichiers :

- ATNV : Attitude nominale
- ATPV : Attitude prédite
- ORHV : orbite centrée sur le soleil (phase croisière)
- ORVV : orbite centrée sur vénus (phase orbite)

Ces fichiers sont mis à jour régulièrement sur le serveur. Un ORVV couvre au maximum la durée d'un mois.

### 5.1.2.Les données venant du SSCWeb

L'orbite des satellites présents au SSCWeb peut être importée. Les données sont mis à jour manuellement via le script \$HOME/i3dv/server/dpc/script/updatefilesscweb.ksh.

Le fichier de configuration est situé dans \$HOME/i3dv\_files/conf/updfilesscweb.conf.

Pour chaque satellite à mettre à jour il faut le déclarer dans le fichier de configuration via une ligne de paramètres. Celui-ci décrit comment renseigner les informations :

```
#1: description
#2: local kernel name
#3: mission name to be retrieved from sscweb
#4: naif id
#5: step in seconds
#6: start date to retrieve
#7: stop date to retrieve
#8: relative path top store file from orbit_files dir. must end with sscweb
#

#id for earth orbiting spacecraft without ID
#If an Earth orbiting spacecraft lacks a DSN identification code, the NAIF ID is derived from the tracking ID assigned to it by
the US Space Command via:
# NAIF ID = -100000 - US Space Command code
#For example US Space Command assigned the code 15427 to the NOAA 9 spacecraft. This code corresponds to the NAIF ID -
115427.
#List of ids: http://nssdc.gsfc.nasa.gov/nmc/SpacecraftQuery.jsp, http://satellitedebris.net/Database/
#
#*****WARNING WHEN ADDING NEW SC, SET preferences in i3dv/server/web/conf/scparams.conf
```

#### Exemple pour la mission cluster :

```
#Update patterns for cluster
#CLUSTER1;cluster1.bsp;cluster1;-183;1200;2000-08-01T00:00:00;2022-01-01T00:00:00;spice/art/sscweb
#CLUSTER2;cluster2.bsp;cluster2;-185;1200;2000-08-01T00:00:00;2022-01-01T00:00:00;spice/art/sscweb
#CLUSTER3;cluster3.bsp;cluster3;-194;1200;2000-08-01T00:00:00;2022-01-01T00:00:00;spice/art/sscweb
#CLUSTER4;cluster4.bsp;cluster4;-196;1200;2000-08-01T00:00:00;2022-01-01T00:00:00;spice/art/sscweb
```

### 5.1.3. Les autres types de donnée

#### 5.1.3.1. Cassini

##### 5.1.3.1.1. Les fichiers d'orbite

Un fichier de base contenant la phase croisière est pré chargé :

Summary for: 971103\_SCEPH\_LPO\_SP0.bsp

Bodies	Start of Interval (ET)	End of Interval (ET)
-82 CASSINI	1997 OCT 15 09:35:42.882	2004 AUG 01 00:00:00.000

Les fichiers d'orbite après Août 2004 correspondants au critère suivant sont récupérés:

[0-9]{6}??\_SK\_[0-9]{5}\_[0-9]{5}.bsp

##### Exemples :

- 060516AP\_SK\_06136\_06145.bsp
- 060522BP\_SK\_06139\_06186.bsp
- 060606AP\_SK\_06157\_06186.bsp
- ...

##### 5.1.3.1.2. Les fichiers d'attitude

Les fichiers d'attitude correspondants au critère suivant sont récupérés :

[0-9]{5}\_[0-9]{5}ra.bc

#### 5.1.3.2. Galileo

##### 5.1.3.2.1. Les fichiers d'orbite

Le fichier galileo\_complete.bsp contient toutes les données d'orbite de la mission.

Ce fichier est la fusion (spkmerge) des fichiers trouvés sur le site de la NAIF :

s970311a.bsp	1989 OCT 19 01:29:33.260	1995 JUL 02 00:01:01.184
s980326a.bsp	1995 JUL 01 00:00:00.000	1998 JAN 01 00:01:03.183
s000131a.bsp	1997 DEC 01 00:01:03.183	2000 FEB 01 20:01:04.184
s030916a.bsp	2000 FEB 01 00:00:00.000	2003 SEP 30 12:00:00.000

Avec le fichier de paramètre suivant :

```
leapseconds_kernel = naif0008.tls
spk_kernel          = galileo_complete.bsp
bodies              = -77
source_spk_kernel  = s000131a.bsp
source_spk_kernel  = s030916a.bsp
source_spk_kernel  = s970311a.bsp
source_spk_kernel  = s980326a.bsp
```

### 5.1.3.3.Ulysses

#### 5.1.3.3.1.Les fichiers d'orbite

Le fichier 901006\_080701\_ulysses.bsp est la fusion (spkmerge) des fichiers trouvés sur le site de la NAIF :

uly_spk_merge1.bsp	1992 JAN 05 01:00:58.184	2008 JUL 01 00:01:04.184
uly_spk_merge2.bsp	1990 OCT 06 19:05:00.000	2006 JAN 01 00:05:00.000

Avec le fichier de paramètre suivant :

```
leapseconds_kernel = naif0008.tls
spk_kernel          = 901006_080701_ulysses.bsp
bodies              = -55
source_spk_kernel  = uly_spk_merge1.bsp
source_spk_kernel  = uly_spk_merge2.bsp
```

### 5.1.3.4.Stereo

#### 5.1.3.4.1.Les fichiers d'orbite

Le fichier Stereo\_complete.bsp est le résultat de la fusion des fichiers suivants Téléchargés sur <http://sohowww.nascom.nasa.gov/solarsoft/stereo/gen/data/spice>

ahead_2006_326_01.depm.bsp	2006 OCT 26 00:01:05.182	2006 NOV 22 16:31:05.182
behind_2006_326_01.depm.bsp	2006 OCT 26 00:01:05.182	2006 NOV 22 16:31:05.182
STA_2006_326_01.epm.bsp	2006 NOV 22 00:01:05.182	2012 JUL 31 00:01:04.183
STB_2006_326_01.epm.bsp	2006 NOV 22 00:01:05.182	2012 JUL 31 00:01:04.183

Avec le fichier de paramètre suivant :

```
leapseconds_kernel= naif0008.tls
spk_kernel          = complete_stereo.bsp
source_spk_kernel  = ahead_2006_326_01.depm.bsp
source_spk_kernel  = behind_2006_326_01.depm.bsp
source_spk_kernel  = STA_2006_326_01.epm.bsp
source_spk_kernel  = STB_2006_326_01.epm.bsp
```

## 5.1.3.5.SOHO

### 5.1.3.5.1.Les fichiers d'orbite

Un seul fichier est utilisé :

soho\_complete.bsp 1995 DEC 05 00:13:01.183      2007 JAN 14 23:49:05.184

Il a été généré de la façon suivante :

- Génération d'un fichier cdf sur le scweb.
- Echantillonnage à 720 secondes.
- Export des variables epoch, x, y, z en GSE via l'outil exportCDF
- Génération des vitesses via une routine qui calcule la vitesse entre deux points (distance/dt). Le premier point est ignoré
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
OUTPUT_SPK_TYPE = 9
OBJECT_ID      = -21
CENTER_ID     = 399
REF_FRAME_NAME = 'GSE'
FRAME_DEF_FILE = 'heliospheric.tf'
PRODUCER_ID   = 'GFI'
DATA_ORDER    = 'epoch x y z vx vy vz'
DATA_DELIMITER = ''
LEAPSECONDS_FILE = 'naif0008.tls'
EPOCH_STR_LENGTH = 23
POLYNOM_DEGREE = 2
```

### 5.1.3.6. Geotail

#### 5.1.3.6.1. Les fichiers d'orbite

Un seul fichier est utilisé :

geotail\_complete.bsp: 1992 JUL 26 00:12:59.183      2010 DEC 31 23:49:05.183

Il a été généré de la façon suivante :

- Génération d'un fichier cdf sur le scweb.
- Echantillonnage à 720 secondes.
- Export des variables epoch, x, y, z en GSE via l'outil exportCDF
- Génération des vitesses via une routine qui calcule la vitesse entre deux points (distance/dt). Le premier point est ignoré
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
OUTPUT_SPK_TYPE = 9
OBJECT_ID      = -1
CENTER_ID     = 399
REF_FRAME_NAME = 'GSE'
FRAME_DEF_FILE = 'heliospheric.tf'
PRODUCER_ID   = 'GFI'
DATA_ORDER    = 'epoch x y z vx vy vz'
DATA_DELIMITER = ''
LEAPSECONDS_FILE = 'naif0008.tls'
EPOCH_STR_LENGTH = 23
POLYNOM_DEGREE = 2
```

### 5.1.3.7. Messenger

#### 5.1.3.7.1. Les fichiers d'orbite

Un seul fichier est utilisé. Il est récupéré sur le site de la naïf :

- msgr\_20040803\_20120401\_od051.bsp:

2004 AUG 03 06:00:20.183      2012 APR 01 10:31:05.185

### 5.1.3.8.INTERBALL

La mission INTERBALL comporte 2 missions : Interball Tail et Interball Aurora.

#### 5.1.3.8.1. Les fichiers d'orbite

interball\_aurora.bsp : -554 1996-09-02T00:10:00.000 → 1999-07-15T07:55:00.000

interball\_tail.bsp : -555 1995-08-04T00:30:00.000 → 2000-10-10T07:45:00.000

Ils ont été générés de la façon suivante :

- Génération de 2 fichiers .txt sur le sscweb
- Echantillonnage : 5 min pour Interball Aurora / 15 min pour Interball Tail
- Fichier de la forme (par ligne) : Epoch X Y Z
- La Terre (ID=399) a été définie comme le centre de l'observation.

Un calcul de vitesse a été effectué pour générer un fichier comprenant les vitesses selon les 3 axes pour, créer le fichier ASCII comme :

```
Epoch X Y Z Vx Vy Vz
```

Pour chaque satellite (OBJECT\_ID=-554, et -555), création du kernel spice via mkspk et le fichier de paramètre suivant :

```
INPUT_DATA_TYPE = 'STATES'  
OUTPUT_SPK_TYPE = 9  
OBJECT_ID      = -554  
CENTER_ID      = 399  
COMMENT_FILE   = 'comment_file.txt'  
IGNORE_FIRST_LINE = 0  
REF_FRAME_NAME = 'J2000'  
INPUT_DATA_UNITS = 'DISTANCES=KM'  
  
FRAME_DEF_FILE = 'heliospheric.tf'  
PRODUCER_ID    = 'Franck Ferreira, GFI Informatique, Toulouse, FRANCE'  
DATA_ORDER     = 'epoch x y z vx vy vz'  
DATA_DELIMITER = ''  
LEAPSECONDS_FILE = 'naif0009.tls'  
EPOCH_STR_LENGTH = 19  
POLYNOM_DEGREE = 8
```

#### 5.1.3.8.2.Les fichiers d'attitude

Aucun fichier d'attitude n'est disponible à ce jour.

## 5.1.3.9.GIOTTO

### 5.1.3.9.1.Les fichiers d'orbite

Deux noyaux (correspondants à l'observation de la comète de HALLEY (1986) et à la comète de GRIGG-SKJELLERUP (1992)) ont été générés pour la mission GIOTTO ; les données entre ces deux observations étant manquantes. Deux fichiers distincts ont donc été générés :

giotto\_1986.bsp : -75 1986-03-05T01:46:39.000 → 1986-03-16T23:06:39.000

giotto\_1992.bsp : -75 1992-01-02T12:00:58.000 → 1992-12-31T11:59:00.000

Ils ont été générés de la façon suivante :

- Récupération d'un fichier ASCII (.txt) depuis le site (giotto 1986)  
[http://pdssbn.astro.umd.edu/volume/hal\\_0025/giotto/ephemris/](http://pdssbn.astro.umd.edu/volume/hal_0025/giotto/ephemris/)
- Récupération d'un fichier ASCII (.txt) depuis le site GSFC de la NASA (giotto 1992)  
<http://cohoweb.gsfc.nasa.gov/helios/heli.html>
- Echantillonnage : 1h15 pour Giotto 1986 & 1 jour pour Giotto 1992
- Fichier de la forme (par ligne) : Epoch X Y Z
- Le soleil (ID=10) a été défini comme le centre de l'observation.

Un calcul de vitesse a été effectué pour générer un fichier comprenant les vitesses selon les 3 axes pour, créer le fichier ASCII comme :

Epoch X Y Z Vx Vy Vz

Pour chaque période (OBJECT\_ID=-75), création du kernel spice via mkspk et le fichier de paramètre suivant :

```
INPUT_DATA_TYPE = 'STATES'
OUTPUT_SPK_TYPE = 9
OBJECT_ID      = -75
CENTER_ID     = 10
COMMENT_FILE  = 'comment_file.txt'
IGNORE_FIRST_LINE = 0
REF_FRAME_NAME = 'ECLIPJ2000'
INPUT_DATA_UNITS = 'DISTANCES=KM'

FRAME_DEF_FILE = 'heliospheric.tf'
PRODUCER_ID   = 'Franck Ferreira, GFI Informatique, Toulouse, FRANCE'
DATA_ORDER    = 'epoch x y z vx vy vz'
DATA_DELIMITER = ''
LEAPSECONDS_FILE = 'naif0009.tls'
EPOCH_STR_LENGTH = 19
POLYNOM_DEGREE = 8
```

### 5.1.3.9.2.Les fichiers d'attitude

Aucun fichier d'attitude n'est disponible à ce jour.

### 5.1.3.10.JASON2

Les fichiers ASCII sont récupérés sur un serveur spécifique de la nasa : cddis.gsfc.nasa.gov.

Les fichiers sont intégrés automatiquement par les procédures de mise à jour.

Les fichiers suivants sont récupérés :

```
/pub/doris/products/orbits/ssa/ja2/ssaja.*
/pub/doris/ancillary/quaternions/ja2/2011/ja2qbody.*
/pub/doris/ancillary/quaternions/ja2/2011/ja2qsolp.*
```

#### 5.1.3.10.1.Les fichiers d'orbite

Le fichier principal se nomme JASON2\_EPHE\_merge.bsp.

Il est issu des fichiers d'orbite ASCII qui doivent contenir le temps au format UTC, la position x,y,z en Km suivi du vecteur d'état.

#### Exemple de fichier :

```
2008-06-22T12:03:12 -4510.977929027 -1203.719445050 5574.617500301 -5.484711246 -1.479205827 -4.750377301
2008-06-22T12:03:44 -4684.069113296 -1250.408340045 5419.673247581 -5.332531824 -1.438592300 -4.932787211
2008-06-22T12:04:16 -4852.199457478 -1295.773030016 5258.974802098 -5.174685717 -1.396450216 -5.109982174
2008-06-22T12:04:48 -5015.190266299 -1339.765305340 5092.692100395 -5.011339248 -1.352823944 -5.281772310
```

Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -553
CENTER_ID = 399
REF_FRAME_NAME = 'IAU_EARTH'
LINES_PER_RECORD = 3
IGNORE_FIRST_LINE = 22
FRAME_DEF_FILE = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/heliospheric.tf'
PRODUCER_ID = 'Laurent BEIGBEDER, GFI Informatique, Toulouse, FRANCE'
DATA_ORDER = 'skip epoch skip x y z skip skip vx vy vz skip'
DATA_DELIMITER = ''
TIME_WRAPPER = '# UTC'
LEAPSECONDS_FILE = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/naif_tai.tls'
INPUT_DATA_UNITS = 'DISTANCES=KM'
EPOCH_STR_LENGTH = 28
POLYNOM_DEGREE = 8
```

Puis fusion des fichiers de mise à jour avec le fichier principal via spkmerge et le fichier de configuration suivant (exemple) :

```
leapseconds_kernel = naif0009.tls
spk_kernel          = JASON2_EPHE_merge.bsp
source_spk_kernel  = JASON2_EPHE_merge_old.bsp
source_spk_kernel  = JASON2_EPHE_x.bsp
```

Ce fichier est généré dynamiquement par les procédures de mise à jour.

### 5.1.3.10.2. Les fichiers d'attitude

Le fichier principal se nomme JASON2\_QUAT\_merge.bsp.

Il est issu des fichiers ASCII qui doivent contenir le temps au format UTC suivi des quaternions.

#### Exemple de fichier :

```
2008-10-12T12:01:26.233 0.535928 0.383699 0.374400 0.652212
2008-10-12T12:02:30.233 0.535928 0.383699 0.374400 0.652212
2008-10-12T12:04:06.233 0.535928 0.383699 0.374400 0.652212
2008-10-12T12:05:10.233 0.535928 0.383699 0.374400 0.652212
```

Création du kernel spice via msopck et le fichier de paramètre suivant :

```
\begindata
LSK_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/naif0009.tls'
SCLK_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/art/jason2.tsc'
COMMENT_FILE   = 'jason2_comment_file.txt'
PRODUCER_ID    = 'Laurent BEIGBEDER, GFI Informatique, Toulouse, FRANCE'
INTERNAL_FILE_NAME = 'internal file name string'
CK_SEGMENT_ID  = 'JASON2 ATT'
CK_TYPE        = 3
INSTRUMENT_ID  = -553000
REFERENCE_FRAME_NAME = 'J2000'
MAXIMUM_VALID_INTERVAL = 86400
INPUT_TIME_TYPE = 'UTC'
INPUT_DATA_TYPE = 'SPICE QUATERNIONS'
ANGULAR_RATE_PRESENT = 'MAKE UP'
```

Puis fusion des fichiers de mise à jour avec le fichier principal via dafcat (concaténation) et cksmrg(fusion des segments) avec les options suivantes :

```
cksmrg -K naif0009.tls corot.tsc corot.tf -I tmp_mergefile.bc -O JASON2_QUAT_merge.bsp
-S 'Segment merged by automatic multimission 3dview dpc process'
-F 'File merged by automatic multimission 3dview dpc process'
-B -553000 -R J2000 -A MAKEAVRG -T 3650 days
```

Ce fichier est généré dynamiquement par les procédures de mise à jour.

### 5.1.3.10.3. Les fichiers d'orientation des générateurs solaires

Le fichier principal se nomme JASON2\_SADM\_merge.bsp.

Il est issu des fichiers ASCII qui doivent contenir le temps au format UTC suivi des angles Z, Y et Z en radians. Comme les panneaux solaire ne tournent qu'autour de Y, seul Y doit être différent de 0 dans le fichier.

#### Exemple de fichier :

```
2008-06-22T12:02:24.678 0.000000 -0.242128 0.000000
2008-06-22T12:05:04.679 0.000000 -0.242128 0.000000
2008-06-22T12:07:44.680 0.000000 -0.242128 0.000000
2008-06-22T12:10:24.681 0.000000 -0.242128 0.000000
```

Création du kernel spice via msopck et le fichier de paramètre suivant :

```
\begindata
LSK_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/naif0009.tls'
SCLK_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/art/jason2.tsc'
FRAMES_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/art/jason2.tf'
PRODUCER_ID = 'Laurent BEIGBEDER, GFI Informatique, Toulouse, FRANCE'
INTERNAL_FILE_NAME = 'JASON2 SA orientation'
CK_SEGMENT_ID = 'JASON SA orientation'
CK_TYPE = 3
INSTRUMENT_ID = -553015
REFERENCE_FRAME_NAME = 'JASON2_SA+Y_ZERO'
DOWN_SAMPLE_TOLERANCE = 0.001
MAXIMUM_VALID_INTERVAL = 172800
INPUT_TIME_TYPE = 'UTC'
INPUT_DATA_TYPE = 'EULER ANGLES'
```

Puis fusion des fichiers de mise à jour avec le fichier principal via dafcat (concaténation) et cksmrg(fusion des segments) avec les options suivantes :

```
cksmrg -K naif0009.tls corot.tsc corot.tf -I tmp_mergefile.bc -O JASON2_SADM_merge.bsp
-S 'Segment merged by automatic multimision 3dview dpc process'
-F 'File merged by automatic multimision 3dview dpc process'
-B -553015 -R COROT_SA+Y_ZERO -A MAKEAVRG -T 30 days
```

Ce fichier est généré dynamiquement par les procédures de mise à jour.

### 5.1.3.11.PICARD

Les fichiers ASCII sont récupérés sur un serveur spécifique: picard.busoc.be.

Les fichiers sont intégrés automatiquement par les procédures de mise à jour.

Les fichiers suivants sont récupérés :

```
orbital_data/geocentric/DETERMINATED.*J2000.*gz
orbital_data/attitude/ R_PIC_ATT.*gz
```

#### 5.1.3.11.1.Les fichiers d'orbite

Le fichier principal se nomme PICARD\_EPHE\_merge.bsp.

Il est issu des fichiers d'orbite ASCII qui doivent contenir le temps au format UTC, la position x,y,z en Km suivi du vecteur d'état.

#### Exemple de fichier :

```
2008-06-22T12:03:12 -4510.977929027 -1203.719445050 5574.617500301 -5.484711246 -1.479205827 -4.750377301
2008-06-22T12:03:44 -4684.069113296 -1250.408340045 5419.673247581 -5.332531824 -1.438592300 -4.932787211
2008-06-22T12:04:16 -4852.199457478 -1295.773030016 5258.974802098 -5.174685717 -1.396450216 -5.109982174
2008-06-22T12:04:48 -5015.190266299 -1339.765305340 5092.692100395 -5.011339248 -1.352823944 -5.281772310
```

Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
OUTPUT_SPK_TYPE = 9
OBJECT_ID      = -556
CENTER_ID      = 399
REF_FRAME_NAME = 'J2000'
FRAME_DEF_FILE = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/heliospheric.tf'
PRODUCER_ID    = 'Laurent BEIGBEDER, GFI Informatique, Toulouse, FRANCE'
DATA_ORDER     = 'epoch x y z vx vy vz'
DATA_DELIMITER = ''
LEAPSECONDS_FILE = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/naif0009.tls'
INPUT_DATA_UNITS = 'DISTANCES=METERS'
EPOCH_STR_LENGTH = 19
POLYNOM_DEGREE  = 8
```

Puis fusion des fichiers de mise à jour avec le fichier principal via spkmerge et le fichier de configuration suivant (exemple) :

```
leapseconds_kernel = naif0009.tls
spk_kernel          = PICARD_EPHE_merge.bsp
source_spk_kernel  = PICARD_EPHE_merge_old.bsp
source_spk_kernel  = PICARD_EPHE_x.bsp
```

Ce fichier est généré dynamiquement par les procédures de mise à jour.

### 5.1.3.11.2. Les fichiers d'attitude

Le fichier principal se nomme PICARD\_QUAT\_merge.bsp.

Il est issu des fichiers ASCII qui doivent contenir le temps au format UTC suivi des quaternions.

#### Exemple de fichier :

```
2008-10-12T12:01:26.233 0.535928 0.383699 0.374400 0.652212
2008-10-12T12:02:30.233 0.535928 0.383699 0.374400 0.652212
2008-10-12T12:04:06.233 0.535928 0.383699 0.374400 0.652212
2008-10-12T12:05:10.233 0.535928 0.383699 0.374400 0.652212
```

Création du kernel spice via msopck et le fichier de paramètre suivant :

```
\begindata
LSK_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/naif0009.tls'
SCLK_FILE_NAME = '/internet2/3DVIEW/server_root/data/orbitfiles/spice/art/picard.tsc'
COMMENT_FILE   = 'picard_comment_file.txt'
PRODUCER_ID    = 'Laurent BEIGBEDER, GFI Informatique, Toulouse, FRANCE'
INTERNAL_FILE_NAME = 'internal file name string'
CK_SEGMENT_ID  = 'PICARD ATT'
CK_TYPE = 3
INSTRUMENT_ID = -556000
REFERENCE_FRAME_NAME = 'J2000'
MAXIMUM_VALID_INTERVAL = 86400
INPUT_TIME_TYPE = 'UTC'
INPUT_DATA_TYPE = 'SPICE QUATERNIONS'
ANGULAR_RATE_PRESENT = 'MAKE UP'
```

Puis fusion des fichiers de mise à jour avec le fichier principal via dafcat (concaténation) et cksmrg(fusion des segments) avec les options suivantes :

```
cksmrg -K naif0009.tls corot.tsc corot.tf -I tmp_mergefile.bc -O PICARD_QUAT_merge.bsp
-S 'Segment merged by automatic multimission 3dview dpc process'
-F 'File merged by automatic multimission 3dview dpc process'
-B -556000 -R J2000 -A MAKEAVRG -T 3650 days
```

Ce fichier est généré dynamiquement par les procédures de mise à jour.

### 5.1.3.12.MRO

Les fichiers ASCII sont récupérés sur le site de la naif.

Les fichiers sont récupérés automatiquement par les procédures de mise à jour.

Les fichiers suivants sont récupérés :

/pub/naif/MRO/kernels/spk/mro\_psp[1-9][0-9].bsp\$

#### 5.1.3.12.1.Les fichiers d'orbite

Les fichiers sont directement récupérés au format SPICE et n'ont pas besoin d'être traités.

#### 5.1.3.12.2.Les fichiers d'attitude

En raison du volume des données d'attitude, elles ne sont pas récupérées.

### 5.1.3.13.Juno

#### 5.1.3.13.1.Les fichiers d'orbite

Un seul fichier est utilisé. Il est récupéré sur le site de la NAIF :

[ftp://naif.jpl.nasa.gov/pub/naif/JUNO/kernels/spk/zzarchive/110805LD-120918-0920DSM-160705AD-to-pj34\\_r090608\\_v2.bsp](ftp://naif.jpl.nasa.gov/pub/naif/JUNO/kernels/spk/zzarchive/110805LD-120918-0920DSM-160705AD-to-pj34_r090608_v2.bsp):

La couverture temporelle est la suivante : 2011-08-05T17:00:02 2017-10-16T21:50:00.

### 5.1.3.14.JUICE

#### 5.1.3.14.1.Les fichiers d'orbite

Un seul fichier est utilisé. Il a été fourni par Nicolas André par mail le 08/08/2012 : mantra.juice\_2022\_ipc\_eur\_inc\_gan\_003.bsp.

La couverture temporelle est la suivante : 2022-06-01T05:47:00; 2033-07-04T18:08:00.

## 5.1.3.15. HELIOS

### 5.1.3.15.1. Les fichiers d'orbite

Un seul fichier est utilisé pour chaque satellite (Helios1 et Helios2). Ils sont disponibles à l'url suivante : <http://naif.jpl.nasa.gov/pub/naif/HELIOS/kernels/spk/>.

La couverture temporelle est la suivante :

Helios 1 : 1974-12-10T07:54:00.000;1981-09-30T00:00:00.000

Helios 2 : 1976-01-15T06:24:00.000;1980-03-09T00:00:00.000

## 5.1.3.16. Pionner

### 5.1.3.16.1. Les fichiers d'orbite

Les fichiers BSP sont récupérés sur le site de la naif :

<http://naif.jpl.nasa.gov/pub/naif/PIONEER10/kernels/spk/p10-a.bsp>

<http://naif.jpl.nasa.gov/pub/naif/PIONEER11/kernels/spk/p11-a.bsp>

[http://naif.jpl.nasa.gov/pub/naif/PIONEER12/kernels/spk/pvo\\_781209\\_820908\\_ssd1999.bsp](http://naif.jpl.nasa.gov/pub/naif/PIONEER12/kernels/spk/pvo_781209_820908_ssd1999.bsp)

[http://naif.jpl.nasa.gov/pub/naif/PIONEER12/kernels/spk/pvo\\_871229\\_880601\\_ssd1999.bsp](http://naif.jpl.nasa.gov/pub/naif/PIONEER12/kernels/spk/pvo_871229_880601_ssd1999.bsp)

## 5.1.3.17. Phobos2

### 5.1.3.17.1. Les fichiers d'orbite

Le fichier d'orbite est récupéré sur le site de la naif :

<http://naif.jpl.nasa.gov/pub/naif/PHOBOS88/kernels/spk/p88mrg.bsp>

## 5.1.3.18. Chandrayan

### 5.1.3.18.1. Les fichiers d'orbite

Un fichier principal couvrant la période 1999 à 2015 est récupéré sur le site de la naif :

[http://naif.jpl.nasa.gov/pub/naif/CHANDRA/kernels/spk/chandra\\_1.bsp](http://naif.jpl.nasa.gov/pub/naif/CHANDRA/kernels/spk/chandra_1.bsp)

Ce fichier est récupéré automatiquement par les procédures de mise à jour.

Un 2<sup>nd</sup> fichier généré à partir du site web Horizon de la Nasa:

chandra\_1\_2015\_2016.bsp

Il a été généré de la façon suivante :

- Génération d'un fichier txt à partir du site Horizon.
- Echantillonnage à 30 minutes
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
INPUT_DATA_UNITS = ('ANGLES=DEGREES' 'DISTANCES=km')
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -151
OBJECT_NAME = 'chandra-1'
CENTER_ID = 10
REF_FRAME_NAME = 'ECLIPJ2000'
PRODUCER_ID = 'GFI'
DATA_ORDER = 'skip epoch x y z vx vy vz'
DATA_DELIMITER = ','
LEAPSECONDS_FILE = 'naif0011.tls'
EPOCH_STR_LENGTH = 30
POLYNOM_DEGREE = 8
IGNORE_FIRST_LINE = 76
```

### 5.1.3.19.DSCOVER

#### 5.1.3.19.1.Les fichiers d'orbite

Un fichier généré à partir du site web Horizon de la Nasa couvrant la période 2015-2017:

DSCOVER.bsp

Il a été généré de la façon suivante :

- Génération d'un fichier txt à partir du site Horizon.
- Echantillonnage à 1 jour
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
INPUT_DATA_UNITS = ('ANGLES=DEGREES' 'DISTANCES=km')
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -79
OBJECT_NAME = 'DSCOVER'
CENTER_ID = 10
REF_FRAME_NAME = 'ECLIPJ2000'
PRODUCER_ID = 'GFI'
DATA_ORDER = 'skip epoch x y z vx vy vz'
DATA_DELIMITER = ','
LEAPSECONDS_FILE = 'naif0011.tls'
EPOCH_STR_LENGTH = 30
POLYNOM_DEGREE = 8
```

## 5.1.3.20.GAIA

### 5.1.3.20.1.Les fichiers d'orbite

Un fichier généré à partir du site web Horizon de la Nasa couvrant la période 2013-2019  
gaia\_complete.bsp

Il a été généré de la façon suivante :

- Génération d'un fichier txt à partir du site Horizon.
- Echantillonnage à 1 jour
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
INPUT_DATA_UNITS = ('ANGLES=DEGREES' 'DISTANCES=km')
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -139479
OBJECT_NAME = 'Gaia'
CENTER_ID = 10
REF_FRAME_NAME = 'ECLIPJ2000'
PRODUCER_ID = 'GFI'
DATA_ORDER = 'skip epoch x y z vx vy vz'
DATA_DELIMITER = ','
LEAPSECONDS_FILE = 'naif0011.tls'
EPOCH_STR_LENGTH = 30
POLYNOM_DEGREE = 8
IGNORE_FIRST_LINE = 76
```

## 5.1.3.21.Herschel

### 5.1.3.21.1.Les fichiers d'orbite

Un fichier généré à partir du site web Horizon de la Nasa couvrant la période 2009-2050  
herschel\_2009\_2050.bsp

Il a été généré de la façon suivante :

- Génération d'un fichier txt à partir du site Horizon.
- Echantillonnage à 1 jour
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
INPUT_DATA_UNITS = ('ANGLES=DEGREES' 'DISTANCES=km')
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -486
OBJECT_NAME = 'Herschel'
CENTER_ID = 10
REF_FRAME_NAME = 'ECLIPJ2000'
PRODUCER_ID = 'GFI'
DATA_ORDER = 'skip epoch x y z vx vy vz'
DATA_DELIMITER = ','
LEAPSECONDS_FILE = 'naif0011.tls'
EPOCH_STR_LENGTH = 30
POLYNOM_DEGREE = 8
IGNORE_FIRST_LINE = 76
```

### 5.1.3.22.Planck

#### 5.1.3.22.1.Les fichiers d'orbite

Un fichier généré à partir du site web Horizon de la Nasa couvrant la période 2009-2050

herschel\_2009\_2050.bsp

Il a été généré de la façon suivante :

- Génération d'un fichier txt à partir du site Horizon.
- Echantillonnage à 1 jour
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
INPUT_DATA_UNITS = ('ANGLES=DEGREES' 'DISTANCES=km')
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -489
OBJECT_NAME = 'Planck'
CENTER_ID = 10
REF_FRAME_NAME = 'ECLIPJ2000'
PRODUCER_ID = 'GFI'
DATA_ORDER = 'skip epoch x y z vx vy vz'
DATA_DELIMITER = ','
LEAPSECONDS_FILE = 'naif0011.tls'
EPOCH_STR_LENGTH = 30
POLYNOM_DEGREE = 8
IGNORE_FIRST_LINE = 76
```

### 5.1.3.23.Xmm

#### 5.1.3.23.1.Les fichiers d'orbite

Un fichier généré à partir du site web Horizon de la Nasa couvrant la période 2000-2015

xmm\_2000\_2015.bsp

Il a été généré de la façon suivante :

- Génération d'un fichier txt à partir du site Horizon.
- Echantillonnage à 30 minutes
- Création du kernel spice via mkspk et le fichier de paramètre suivant :

```
\begindata
INPUT_DATA_TYPE = 'STATES'
INPUT_DATA_UNITS = ('ANGLES=DEGREES' 'DISTANCES=km')
OUTPUT_SPK_TYPE = 9
OBJECT_ID = -125989
OBJECT_NAME = 'XMM'
CENTER_ID = 10
REF_FRAME_NAME = 'ECLIPJ2000'
PRODUCER_ID = 'GFI'
DATA_ORDER = 'skip epoch x y z vx vy vz'
DATA_DELIMITER = ','
LEAPSECONDS_FILE = 'naif0011.tls'
EPOCH_STR_LENGTH = 30
POLYNOM_DEGREE = 8
IGNORE_FIRST_LINE = 76
```

## 5.2.LES SERVEURS

Les données sont récupérées sur des serveurs à accès public. Soit les spice kernels sont disponible directement sur le site de la Naif, soit ils sont générés à partir de fichiers ASCII des positions et des vitesses de la sonde via l'outil spice *mkspk*. Les fichiers ascii sont généralement obtenus en faisant des requêtes manuelles sur le site du sscweb s'ils ne sont pas mis à jour via des sites FTP spécifiques (COROT, PICARD, ...).

Depuis le projet Impex, une méthode de génération de kernels avec une connexion directe au sscweb a été ajoutée.

Seuls les principaux serveurs sont décrits ici.

### 5.2.1.Le serveur NAIF

Le serveur NAIF (NASA's Navigation and Ancillary Information Facility) est le serveur contenant les spice kernels. Son adresse est la suivante : *ftp://naif.jpl.nasa.gov*.

Le site *http://naif.jpl.nasa.gov/naif/data.html* permet de savoir ce que contient le serveur ftp.

### 5.2.2.Le rapatriement des fichiers

Pour mettre à jour les fichiers de données du logiciel, des applications java sont utilisées.

Elles permettent de rapatrier les fichiers sur des serveurs FTP, de les convertir si nécessaire, et de mettre à jour les différents index.

### 5.2.3.Les fichiers scripts

Ils se situent dans le répertoire `$HOME/m3dv_files/apps/scripts`.

Pour mettre à jour les fichiers d'orbite, il suffit de lancer le script `update_files.ksh`.

Un script `setpassword.ksh` sert à créer les login/password pour les connexions aux sites ftp. On le lance de la manière suivante : Usage: `setpassword.ksh <ftpid>`

Ou `<ftpid>` est l'identifiant utilisé dans le fichier de configuration des mises à jour décrit plus loin.

Enfin, le script `updatecoverage.ksh` permet de mettre à jour le fichier de couverture des données à partir de l'index de l'ensemble des fichiers.

## 5.2.4. Les fichiers de configuration

Il y a deux sortes de fichiers de configuration pour les mises à jour :

- Un principal listant les missions à mettre à jour
- Des fichiers de conversion pour les missions qui en ont besoin.

Le fichier principal est dans apps/conf/updfile.conf. Il contient des lignes permettant de définir une mise à jour d'un type de fichier pour une mission sur un serveur FTP. Il a le format suivant :

```
<titre>;<id>;<naifid>;<serveur FTP>;DIRECT|PACTE;<ftpid>;<rep distant>;<rep local>;<fichiers à récupérer(regex)>;<Type
fichier>;<type tri>;<programme de conversion>
```

Les autres fichiers permettent aux outils spice de convertir les données en kernel. Ils sont situés dans le répertoire apps/conf/spice.

## 5.2.5. Les missions mises à jour

### 5.2.5.1. Cassini

Ils sont récupérés sur le site de la NAIF (naif.jpl.nasa.gov).

#### 5.2.5.1.1. Fichiers d'orbite

La ligne de configuration est la suivante :

```
CASSINI ORBIT FILES;cass;-82;naif.jpl.nasa.gov;PACTE;naif;/pub/naif/CASSINI/kernels/spk;spice/art/cass;[0-9]{6}.._SK_[0-9]{5}_[0-9]{5}.bsp$;ORBIT;DESC
```

Seront donc récupérés les fichiers correspondant au pattern "[0-9]{6}??\_SK\_[0-9]{5}\_[0-9]{5}.bsp" dans le répertoire `/pub/naif/CASSINI/kernels/spk` et ne se trouvant pas déjà dans le répertoire `%MM3DVIEW_ORBITO_DIR%\spice\art\cass`.

Les fichiers SK contiennent uniquement les données d'orbite de Cassini.

## 5.2.5.2. Rosetta

Les fichiers sont récupérés sur le site de la NAIF ([naif.jpl.nasa.gov](http://naif.jpl.nasa.gov)).

### 5.2.5.2.1. Les fichiers d'orbite

La ligne de configuration est la suivante :

ROSETTA	ORBIT	FILES;ros;-
226;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/ROSETTA/kernels/spk;spice/art/ros;ROR.*BSP\$;ORBIT;ESA		

Les fichiers BSP OR\*R seront récupérés. Ces fichiers permettent de tracer l'orbite de Rosetta.

### 5.2.5.2.2. Les fichiers d'attitude

La ligne de configuration est la suivante :

ROSETTA	ATTITUDE	FILES;ros;-
226000;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/ROSETTA/kernels/ck;spice/art/ros;RAT.*BC\$;ATTITUDE;ESA		

Les fichiers AT\*R seront récupérés. Ces fichiers permettent de d'obtenir l'orientation de Rosetta.

### 5.2.5.2.3. Les fichiers d'orientation des panneaux solaires

La ligne de configuration est la suivante :

ROSETTA	SOLAR	ARRAY	FILES;ros;-
226015;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/ROSETTA/kernels/ck;spice/art/ros;ROS_SA.*BC\$;SOLAR_ARRAY;DESC			

Les fichiers ROS\_SA\* seront récupérés. Ces fichiers permettent d'orienter les panneaux solaires de Rosetta.

## 5.2.5.3. Venus-Express

Les fichiers sont récupérés sur le site de la NAIF ([naif.jpl.nasa.gov](http://naif.jpl.nasa.gov)).

### 5.2.5.3.1. Les fichiers d'orbite

La ligne de configuration est la suivante :

VEX ORBIT	FILES;vex;-248;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/VEX/kernels/spk;spice/art/vex;OR.V.*BSP\$;ORBIT;ESA
-----------	--

Les fichiers OR\*V seront récupérés. Ces fichiers permettent de tracer l'orbite de Venus-Express.

### 5.2.5.3.2. Les fichiers d'attitude

La ligne de configuration est la suivante :

VEX	ATTITUDE	FILES;vex;-
248000;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/VEX/kernels/ck;spice/art/vex;ATNV_.*BC\$;ATTITUDE;ESA		

Les fichiers ATNV seront récupérés. Ces fichiers permettent de d'orienter Venus-Express.

### 5.2.5.3.3. Les fichiers d'orientation des panneaux solaires

La ligne de configuration est la suivante :

VEX	SOLAR	ARRAY	FILES;vex;-
248011;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/VEX/kernels/ck;spice/art/vex;VEX_SA.*BC\$;SOLAR_ARRAY;DESC			

Les fichiers VEX\_SA\* seront récupérés. Ces fichiers permettent d'orienter les panneaux solaires de Venus-Express.

## 5.2.5.4. Mars-Express

Les fichiers sont récupérés sur le site de la NAIF (naif.jpl.nasa.gov).

### 5.2.5.4.1. Les fichiers d'orbite

La ligne de configuration est la suivante :

MEX ORBIT	FILES;mex;-41;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/MEX/kernels/spk;spice/art/mex;OR.M.*BSP\$;ORBIT;ESA
-----------	---

Les fichiers OR\*M seront récupérés. Ces fichiers permettent de tracer l'orbite de Mars-Express.

### 5.2.5.4.2. Les fichiers d'attitude

La ligne de configuration est la suivante :

MEX	ATTITUDE	FILES;mex;-41001;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/MEX/kernels/ck;spice/art/mex;ATNM_P[0-9].*BC\$;ATTITUDE;ESA
-----	----------	--

Les fichiers ATNM seront récupérés. Ces fichiers permettent de d'orienter Mars-Express.

### 5.2.5.4.3. Les fichiers d'orientation des panneaux solaires

La ligne de configuration est la suivante :

MEX 41011;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/MEX/kernels/ck;spice/art/mex;MEX_SA.*BC\$;SOLAR_ARRAY;DESC	SOLAR	ARRAY	FILES;mex;-
---	-------	-------	-------------

Les fichiers MEX\_SA\* seront récupérés. Ces fichiers permettent d'orienter les panneaux solaires de Mars-Express.

### 5.2.5.5. MRO

Seuls les fichiers d'orbite sont récupérés, les fichiers d'attitude étant trop volumineux.

La ligne de configuration est la suivante :

MRO	ORBIT	FILES;mro;-74;naif.jpl.nasa.gov;DIRECT;naif;/pub/naif/MRO/kernels/spk;spice/art/mro;mro_psp[1-9][0-9].bsp\$;ORBIT;DESC
-----	-------	--

Les fichiers sont directement rapatriés au format BSP et n'ont pas besoin d'être transformés.

Chaque fichier contient 3 mois de données.

## 5.2.6. Fichiers d'index

Il est situé par défaut dans le répertoire *\$HOME/i3dv\_files/orbitfiles*.

Voici la liste et leur fonction :

- *naiffiles.lst*: indexe les fichiers d'orbite et d'attitude des missions
- *sccoverage.lst*: couverture min et max des données d'une mission

### 5.2.6.1. naiffiles.lst

Les fichiers d'orbite et d'attitude sont listés dans le fichier d'index *naiffiles.lst*

Ce fichier sert à la recherche des bons fichiers d'orbite et d'attitude lors de la sélection des données via le navigateur.

Il est formaté comme suit :

Id naif; Nom fichier d'orbite; temps début; temps fin

**Exemple :**

-82;040506BP\_OPK\_04122\_08222.bsp;2004-05-01T12:00:00;2008-08-09T00:00:00  
-55;uly\_spk\_merge1-2.bsp;1990-10-06T19:05:00;2008-07-01T00:01:00

**5.2.6.2.sccoverage.lst**

Liste la couverture de données par ID d'instrument.

Il est utilisé pour renseigner l'utilisateur lorsqu'il fait une sélection en dehors de la couverture ou lors de l'affichage du catalogue des données.

Il est formaté comme suit :

Id naif; temps début; temps fin

**Exemple :**

-551015;2006-12-27T15:20:21;2009-05-27T07:19:00  
-551000;2007-04-03T00:00:20;2009-04-20T11:59:45  
-551;2006-12-27T15:13:36.000;2009-05-04T12:00:00.000

## 6. SELECTION ET AFFICHAGE DES DONNEES

### 6.1. GENERALITES

Cette partie présente les deux composantes logicielles principales du CDPP 3DView :

- L'application client 3D qui est l'unique IHM utilisée par le client
- Le serveur fournissant des web services utilisés par le client 3D

### 6.2. LE CLIENT 3D

#### 6.2.1. Lancement

Le lancement se fait via "java web start" qui permet de lancer des applications à distance.

Pour ce lancement, un servlet génère un fichier de données dans un espace temporaire du serveur et le compresse.

Puis, il génère un fichier JNLP (Java Net Launching Protocol) qui permet de lancer l'application CDPP 3Dview. En voici un exemple :

```
<jnlp spec="1.0+" codebase="http://odyssey:8081/cdppservices/client3d/">
<information>
<title>3DView/CDPP odyssey</title>
<vendor>GFI Toulouse for CNES</vendor>
<icon kind="shortcut" href="http://odyssey:8081/cdppservices/images/cdpp3dview_logo_64.ico" width="64" height="64"/>
<icon kind="shortcut" href="http://odyssey:8081/cdppservices/images/cdpp3dview_logo_48.ico" width="48" height="48"/>
<icon kind="shortcut" href="http://odyssey:8081/cdppservices/images/cdpp3dview_logo_16.ico" width="16" height="16"/>
<icon kind="splash" href="http://odyssey:8081/cdppservices/images/cdpp3dview_splash.jpg" height="600" width="600"/>
<offline-allowed/>
<shortcut online="true">
<desktop/>
<menu submenu="3DView"/>
</shortcut>
</information>
<security>
<all-permissions/>
</security>

<resources>
<j2se version="1.6+" max-heap-size="512m"/>
<jar href="jnlp-bootstrap-s.jar" />
<jar href="cdpp3dview-s.jar" part="m3dv" download="lazy"/>
<jar href="jfreechart-s.jar" part="jfc" download="lazy"/>
<jar href="jcommon-s.jar" part="jcom" download="lazy"/>
<jar href="jmfAll-s.jar" part="jmf" download="lazy"/>
<jar href="j3dvrm1-s.jar" part="lvrml" download="lazy"/>
<jar href="j3d3ds-s.jar" part="l3ds" download="lazy"/>
```

```

<jar href="ncCore-s.jar" part="netcdf" download="lazy"/>
<property name="java.util.logging.config.file" value="c3dvClientLogging.properties"/>
<property name="texturesBaseUrl" value="http://odyssey:8081/cdppservices/images/textures"/>
<property name="modelsBaseUrl" value="http://odyssey:8081/cdppservices/models"/>
<property name="wsdlUrl" value="http://odyssey:8081/cdppservices/CdppServices?wsdl"/>
<property name="serverBaseUrl" value="http://odyssey:8081/cdppservices"/>
<property name="gfi.tls.mm3dview.toLoad" value="I3ds m3dv jfc jcom jmf lvrml netcdf j3d"/>
<extension href="http://odyssey:8081/cdppservices/GenAxisJnlp"/>
<property name="gfi.tls.mm3dview.classToLaunch" value="gfi.tls.mm3dview.component.commonmenu.JMulti3DViewControler"/>
</resources>

<resources os="Windows" arch="x86">
  <jar href="j3d/windows-i586-j3d-s.jar" part="j3d" download="lazy"/>
  <property name="j3d.rend" value="ogl"/>
</resources>
<resources os="Windows" arch="amd64">

  <jar href="j3d/windows-amd64-j3d-s.jar" part="j3d" download="lazy"/>
</resources>
<resources os="Linux" arch="i386" >
<nativelib href="j3d/linux-i586-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="Linux" arch="x86" >
<nativelib href="j3d/linux-i586-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="Linux" arch="amd64" >
<nativelib href="j3d/linux-amd64-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="Linux" arch="x86_64" >
<nativelib href="j3d/linux-amd64-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="SunOS" arch="sparc" >
<nativelib href="j3d/solaris-sparc-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="SunOS" arch="sparcv9" >
<nativelib href="j3d/solaris-sparc-v9-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="SunOS" arch="x86" >
<nativelib href="j3d/solaris-x86-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="SunOS" arch="amd64" >
<nativelib href="j3d/solaris-x86-amd64-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="SunOS" arch="x86_64" >
<nativelib href="j3d/solaris-x86-amd64-j3d-s.jar" part="j3d" download="lazy" />
</resources>
<resources os="Mac OS X">
  <jar href="j3d/macosex-universal-j3d-s.jar" part="j3d" download="lazy"/>
  <property name="j3d.rend" value="jogl"/>
</resources>
<application-desc main-class="gfi.tls.mm3dview.component.commonmenu.JMulti3DViewControler"/>
</jnlp>

```

On peut voir dans ce fichier que l'on lance l'application cdp3dview-s.jar qui requiert :

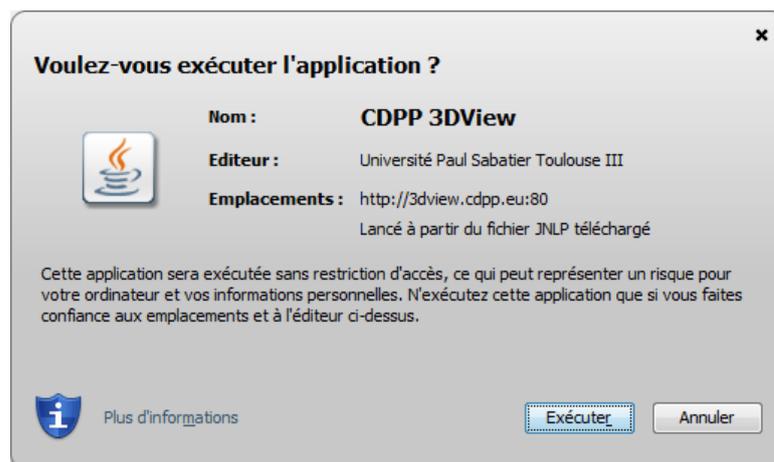
- Une version du JRE 1.7\_045 ou plus,
- Des packages standards signés (\*-s.jar),
- Des ressources spécifiques 3D

Le package *jmfAll-s.jar* est l'API JavaMediaFramework qui permet de générer des films. Ce jar est généré spécialement pour l'application CDPP 3Dview pour ne contenir que le strict minimum nécessaire à son utilisation par l'application.

Les packages java sont signés car les fonctions du CDPP 3Dview nécessitent un accès au disque dur et donc des droits spéciaux. Le certificat qui a permis de signer les applets a été généré librement pour l'application sans utiliser de certificat d'une autorité reconnue. C'est pourquoi, une fenêtre apparaît au lancement de l'application pour signaler que la signature n'a pas été reconnue par un organisme officiel :



Pour la version disponible sur le serveur [3dview.cdpp.eu](http://3dview.cdpp.eu), les jars sont signés manuellement par l'IRAP avec un certificat de l'université Paul Sabatier. Ainsi, sur ce serveur, le message est le suivant :



## 6.2.2. Architecture de l'application Java

L'affichage 3D est réalisé par une application graphique Java s'appuyant principalement sur les classes Swing et le framework java3D.

### 6.2.2.1. Liste des principaux composants utilisés

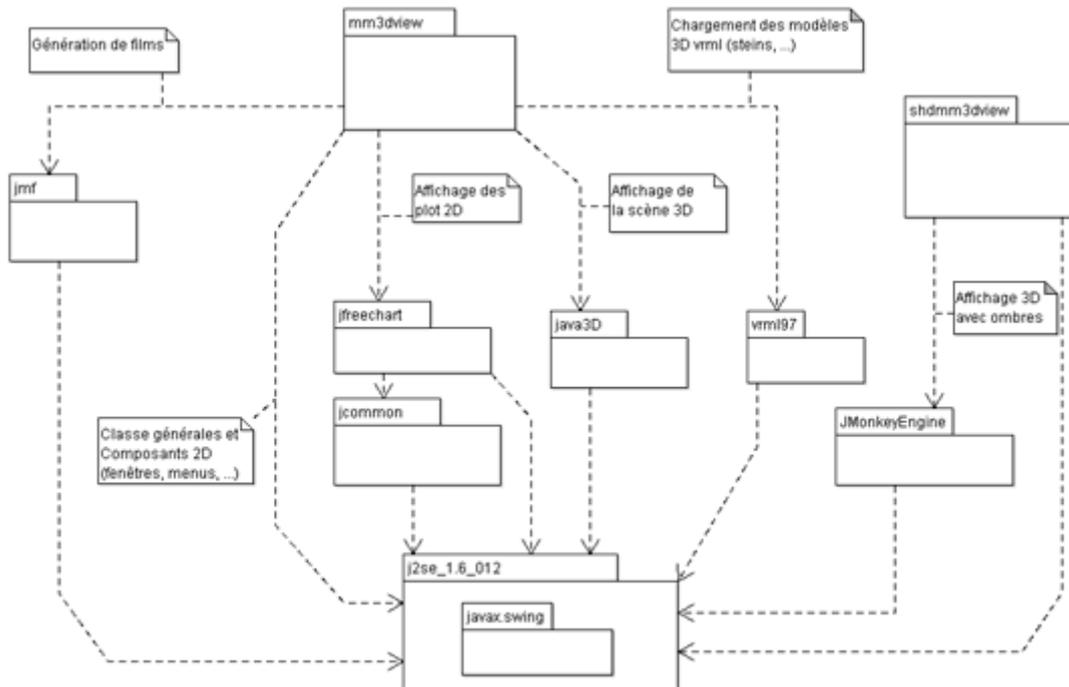
Ces composants sont les suivants :

Nom	Version	Fonctions
Sun Netbeans	8.02	IDE
Java 2 standard Edition	1.8.0_45 1.6.0_45	Parties client 3D Parties serveur
Java Media Framework	2.1.1e	Génération de film à partir de jpeg
Java3D	1.6	Afficher les objets 3D
Jogl	2.1.2	Librairie d'accès à la carte graphique utilisée par java3D
JCommon	1.0.13	Package nécessaire à JfreeChart
JFreeChart	1.0.10	Affichage des plots 2D
NetCDF-Java	4.1	API de lecture de fichiers NetCDF
Jsamp	1.3.2	Utilisée pour se connecter à un hub SAMP
Sgp4	1.0.4	API de lecture des fichiers TLE
CdasWebServicesSdk	1.7.10.12	Client RPC pour l'accès aux services cdaweb
Okhttp	3.2.0	Utiliser pour communiquer avec le Cluster Science Archive par requête https
Okio	1.7.0	Package nécessaire à okhttp

### 6.2.2.2. Les composants

#### 6.2.2.2.1. Dépendances

Le schéma suivant indique le sens et les raisons des dépendances des packages java :



#### 6.2.2.2.2.L'API Java2 Standard Edition

L'API J2SE fournit un ensemble de classes de base Java telle que des listes, des objets graphiques, ... La version 7 permet des améliorations de codage avec des boucles for simplifiées, des listes typées ...

L'API standard contient le package Swing.

Les classes Swing sont disponibles depuis l'apparition de Java2. Elles permettent d'afficher des composants graphiques tels que applets, frames, boîtes de dialogues ...

Si l'application a été développée avec le JDK (Java Development Kit), elle n'a besoin que du JRE (Java Runtime Environment) pour être exécutée. Il est conseillé d'utiliser le dernier JRE **1.8**.

Le JRE est disponible à l'adresse suivante :

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

#### 6.2.2.2.3.L'API java3D

L'API Java3D permet de manipuler et d'afficher des objets géométriques en trois dimensions.

La version 1.6 a été choisie car elle est la version la plus récente. Elle est la seule compatible avec la nouvelle version JAVA7 installé obligatoirement sur MacOSX.

L'API est téléchargée depuis le site 3DView via java web start lorsque l'affichage 3d est lancé si elle n'a pas déjà été téléchargée lors d'un précédent lancement (La taille du package est environ de 3 Mo).

Tous les détails sur l'API se trouvent à l'adresse <https://java3d.dev.java.net/>

#### 6.2.2.2.4.L'API Java Media Framework

Cette API sert à générer les films au format quicktime. Cette API permet au logiciel de concaténer une série d'image pour en faire un film. En effet, pour générer un film, d'abord, toutes les images sont générées, puis, un film au format quicktime est créé en via un flux d'images au format JPEG.

Tous les détails sur l'API se trouvent à l'adresse :

<http://www.oracle.com/technetwork/java/javase/tech/jmf-140515.html>

#### 6.2.2.2.5.L'API JFreeChart

Cette API permet d'afficher des graphiques 2D de différents types : courbes, histogrammes, camemberts, ...

Tous les détails sur l'API se trouvent à l'adresse <http://www.jfree.org/jfreechart>

#### 6.2.2.2.6.L'API VrmI97

Le loader de modèles 3D VRML n'est plus utilisé actuellement mais il a été conservé pour une possible utilisation future. Il était utilisé pour charger les modèles de planètes tels que Steins. Cette API a été développé par la communauté JAVA. Le site de référence se trouve à l'adresse <https://j3d-vrml97.dev.java.net/>.

#### 6.2.2.2.7.L'API 3ds

Le loader de modèles 3DS est utilisé pour charger des modèles de satellites et d'astéroïdes dans la vue générale. Cette API date de 2001 et n'est plus maintenue. Le site de référence se trouve à l'adresse <http://www.starfireresearch.com/services/java3d/inspector3ds.html>.

#### 6.2.2.2.8.L'API netCDF-JAVA

Cette API permet de lire les données sciences récupérées depuis le serveur AMDA.

Elle est créée et maintenue par unidata : <http://www.unidata.ucar.edu/software/netcdf-java/>

#### 6.2.2.2.9.JSAMP

Cette librairie est une implémentation du protocole Simple Application Messaging Protocol (<http://wiki.ivoa.net/twiki/bin/view/IVOA/SampInfo>). Elle fournit des IHM ainsi qu'un hub et permet de ne développer que très peu de code. Elle est utilisée pour recevoir des données ou en envoyer à AMDA par exemple. Elle est disponible à l'adresse <http://www.star.bris.ac.uk/~mbt/jsamp/>

### 6.2.2.2.10.SGP4

Cette API permet de lire les fichiers d'orbites des satellites de la terre au format TLE en utilisant le modèle de perturbations simplifié SGP4 et ses dérivés (SGP, SDP4, SGP8 et SDP8). La description de ce format est disponible à l'adresse <https://celestrak.com/NORAD/documentation/tle-fmt.asp>.

### 6.2.2.2.11.CDAS Web Services Developer's Kit.

Le SDK du cdaweb permet d'accéder par web service aux données du cdaweb. C'est un client web service RPC dont la page est à l'adresse <http://cdaweb.gsfc.nasa.gov/WebServices/DevelopersKit.html>

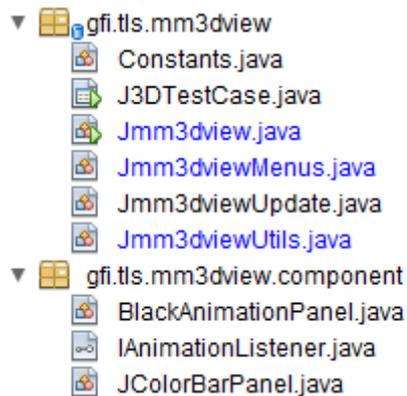
### 6.2.2.2.12.OkHttp

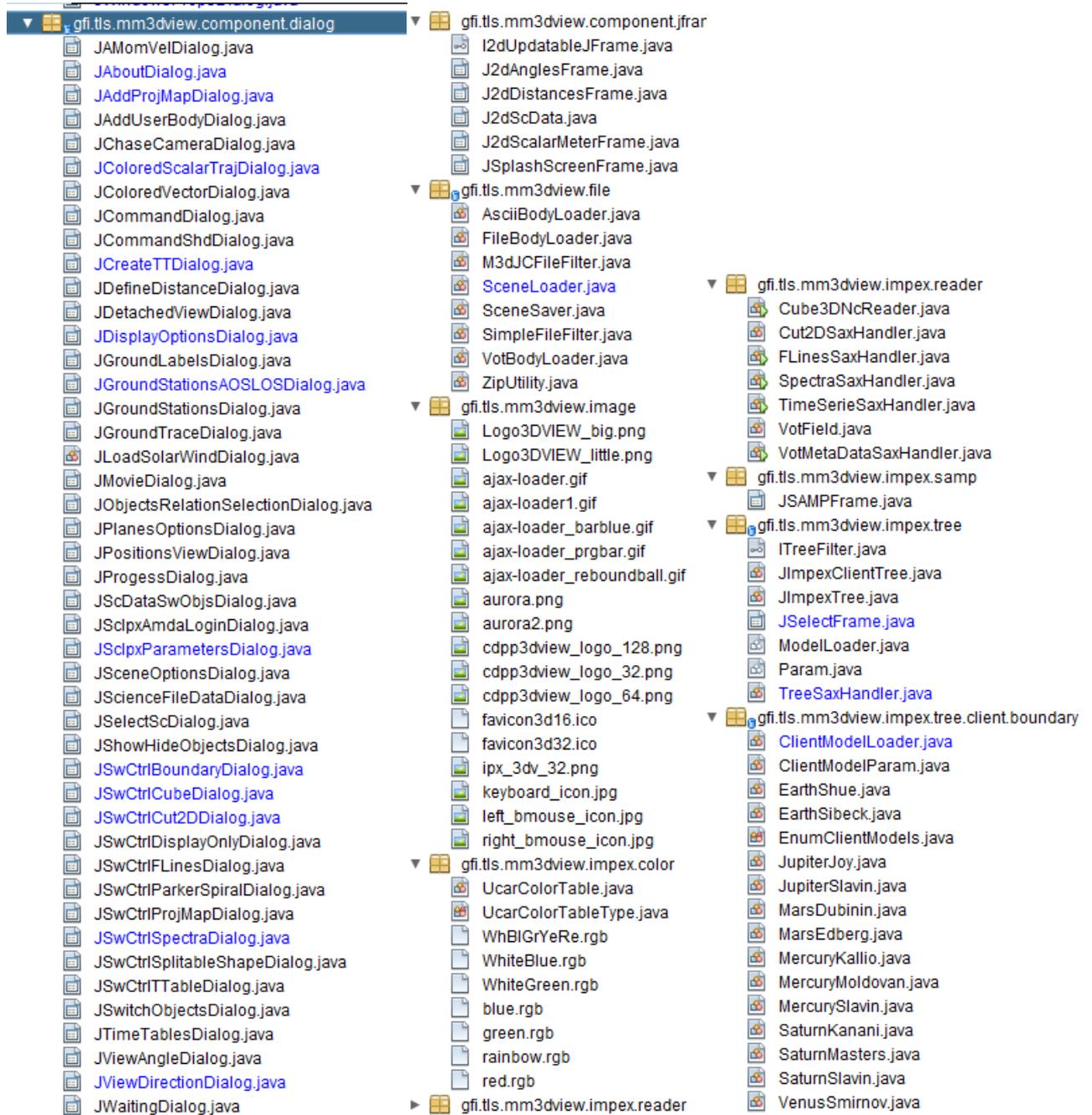
OkHttp est une librairie permettant de dialoguer facilement par requête directe en http. Elle est utilisée pour accéder au Cluster Science Archive. La librairie est disponible ici : <http://square.github.io/okhttp/>

## 6.2.2.3.L'arborescence du package mm3dview

Le package s'appelle mm3dview du fait de l'historique de développement du projet.

Voici le contenu du package client mm3dview développé pour afficher en 3D les données.





<ul style="list-style-type: none"> <li>▼  gfi.tls.mm3dview.impex.color <ul style="list-style-type: none"> <li> UcarColorTable.java</li> <li> UcarColorTableType.java</li> <li> WhBIGrYeRe.rgb</li> <li> WhiteBlue.rgb</li> <li> WhiteGreen.rgb</li> <li> blue.rgb</li> <li> green.rgb</li> <li> rainbow.rgb</li> <li> red.rgb</li> </ul> </li> <li>▼  gfi.tls.mm3dview.impex.reader <ul style="list-style-type: none"> <li> Cube3DNcReader.java</li> <li> Cut2DSaxHandler.java</li> <li> FLinesSaxHandler.java</li> <li> SpectraSaxHandler.java</li> <li> TimeSerieSaxHandler.java</li> <li> VotField.java</li> <li> VotMetaDataSaxHandler.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.impex.samp <ul style="list-style-type: none"> <li> JSAMPFrame.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.impex.tree <ul style="list-style-type: none"> <li> ITreeFilter.java</li> <li> JImpexClientTree.java</li> <li> JImpexTree.java</li> <li> JSelectFrame.java</li> <li> ModelLoader.java</li> <li> Param.java</li> <li> TreeSaxHandler.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.impex.tree.client.boundary <ul style="list-style-type: none"> <li> ClientModelLoader.java</li> <li> ClientModelParam.java</li> <li> EarthShue.java</li> <li> EarthSibeck.java</li> <li> EnumClientModels.java</li> <li> JupiterJoy.java</li> <li> JupiterSlavin.java</li> <li> MarsDubinin.java</li> <li> MarsEdberg.java</li> <li> MercuryKallio.java</li> <li> MercuryMoldovan.java</li> <li> MercurySlavin.java</li> <li> SaturnKanani.java</li> <li> SaturnMasters.java</li> <li> SaturnSlavin.java</li> <li> VenusSmirmov.java</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▼  gfi.tls.mm3dview.impex.tree.model <ul style="list-style-type: none"> <li> CheckBoxNode.java</li> <li> CheckBoxNodeType.java</li> <li> ITooltipNode.java</li> <li> JCheckTree.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.impex.tree.obs <ul style="list-style-type: none"> <li> AmdaDataTreeSaxHandler.java</li> <li> CdawebDataTreeLoader.java</li> <li> Datacenter.java</li> <li> Dataset.java</li> <li> Instrument.java</li> <li> Mission.java</li> <li> ObsModelLoader.java</li> <li> ObsParam.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.impex.tree.sim <ul style="list-style-type: none"> <li> FieldLineConfig.java</li> <li> InputField.java</li> <li> InputPopulation.java</li> <li> JClientBoundaryPanel.java</li> <li> JWSCalcEarthPanel.java</li> <li> JWSCalcOtherPanel.java</li> <li> JWSInputCalcPanel.java</li> <li> JWSInputFieldLinePanel.java</li> <li> JWSInputGetPanel.java</li> <li> NumOut.java</li> <li> ParamPart.java</li> <li> SimModel.java</li> <li> SimModelLoader.java</li> <li> SimParam.java</li> <li> SimRun.java</li> <li> SimRunDomain.java</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▼  gfi.tls.mm3dview.scfile <ul style="list-style-type: none"> <li> ComputationTools.java</li> <li> IScFile.java</li> <li> ScData.java</li> <li> ScFileCDAWebText.java</li> <li> ScFileFlat.java</li> <li> ScFileFlatFilter.java</li> <li> ScFileNetcdf.java</li> <li> ScFileNetcdfFilter.java</li> <li> ScLoader.java</li> <li> ScVariable.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.shape <ul style="list-style-type: none"> <li> AngleObj.java</li> <li> ColoredScalarTraj.java</li> <li> ColoredVectors.java</li> <li> Cube3DShape.java</li> <li> Cut2DShape.java</li> <li> FieldLines.java</li> <li> FieldPoint.java</li> <li> FieldVector.java</li> <li> GroundLabel.java</li> <li> GroundStation.java</li> <li> GroundTrace.java</li> <li> ISwitchableObj.java</li> <li> ProjectionMapShape.java</li> <li> SpectraOnTraj.java</li> <li> SwitchableObj.java</li> <li> TimeTable.java</li> </ul> </li> <li>▼  gfi.tls.mm3dview.shape.body <ul style="list-style-type: none"> <li> Body.java</li> <li> BodyLink.java</li> <li> BodyPartShapes.java</li> <li> BodyShapes.java</li> <li> BodyType.java</li> <li> IBodyLoader.java</li> <li> Instrument.java</li> <li> InstrumentDisplay.java</li> <li> InstrumentRepType.java</li> <li> InstrumentVisibleFrame.java</li> <li> NaifId.java</li> <li> Planet3D.java</li> <li> TargetView.java</li> </ul> </li> </ul>
--	--	---

- ▼  gfi.tls.mm3dview.shape.model
  -  Aurora.java
  -  IModelShape.java
  -  ModelShapeType.java
  -  ParkerSpiral.java
  -  RadiationBelts.java
  -  SAACollisionDetector.java
  -  ShapeCutMode.java
  -  Shock.java
  -  SouthAtlanticAnomaly.java
  -  VanAllenBelt.java
- ▼  gfi.tls.mm3dview.util
  -  AOSLOS.java
  -  ColorTable.java
  -  DateInputVerifier.java
  -  DrawUtil.java
  -  FloatInputVerifier.java
  -  GeomUtil.java
  -  IColorTable.java
  -  IntegerInputVerifier.java
  -  JpegImagesToMovie.java
  -  MouseMove3DListener.java
  -  MovieGenerator.java
  -  RainbowColorTable.java
  -  Star.java
  -  gouraud.frag
  -  gouraud.vert
  -  phong.frag
  -  phong.vert
- ▼  gfi.tls.mm3dview.ws
  -  AmdaWsDataset.java
  -  AmdaWsIndex.java
  -  BodyData.java
  -  ScienceDataType.java
  -  WSUtils.java
  -  WsBodiesLoader.java
- ▼  gfi.tls.mm3dview.ws.ihm
  -  InputSave.java
  -  JAddObjFrame.java
  -  JFileSelectDialog.java
  -  JListScFilesDialog.java
- ▼  gfi.tls.mm3dview.ws.ihm.model
  -  DateLabelFormatter.java
  -  JButtonTableCellEditor.java
  -  JButtonTableCellRenderer.java
  -  JTableRadioButtonCellEditor.java
  -  JTableRadioButtonCellRenderer.java
- ▼  jp.testj3d
  -  J3dOffscreenTestCase.java

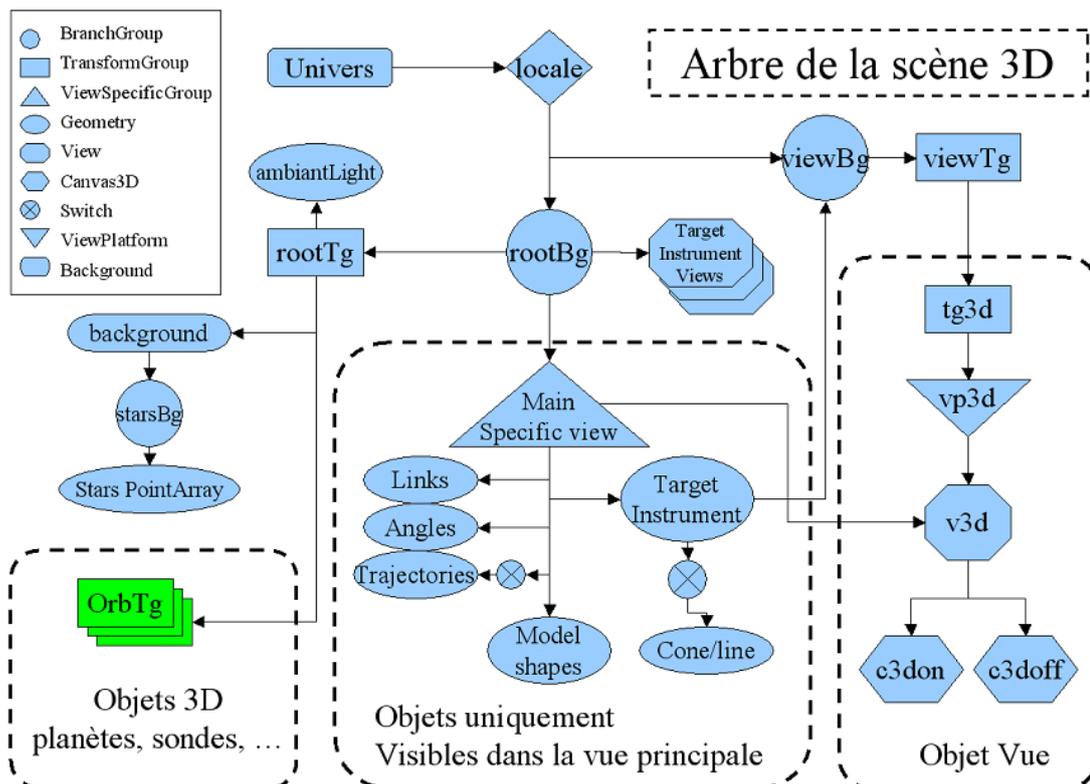
### 6.2.3. Implémentation des fonctionnalités

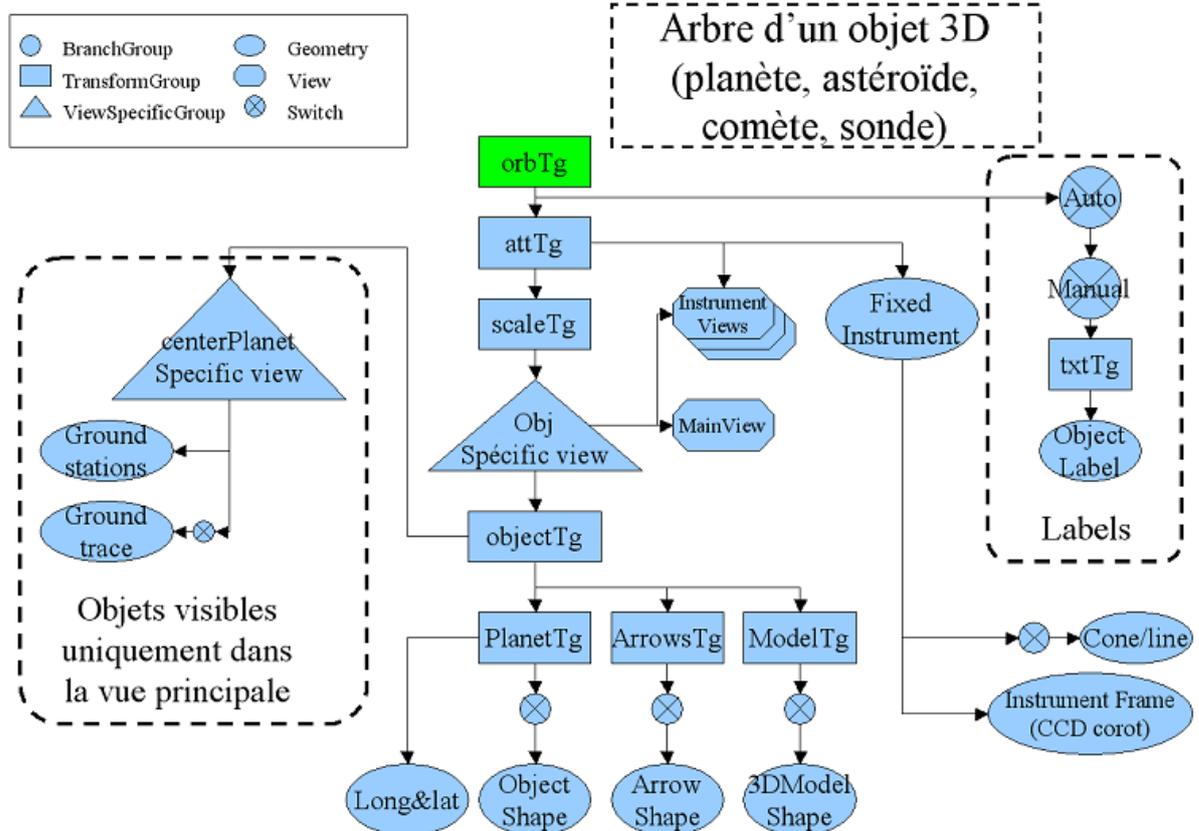
Le plot 3d permet outre l'affichage 3D :

- De simuler l'éclairage du soleil sur les corps
- De simuler l'orientation d'instruments
- D'exporter les données au format ASCII
- De sauvegarder des images
- De générer des films
- D'afficher l'évolution de distances et d'angles
- D'afficher des modèles d'ondes de choc et de magnétopause
- D'afficher des modèles physiques analytiques intégrés
- De récupérer des données scientifiques sur un serveur distant et de les afficher.
- D'afficher les étoiles

#### 6.2.3.1. Composition de la scène 3D

JAVA3D et Jmonkey Engine composent des scènes 3D structurées en arbre. La scène 3D de l'application des ombres étant très simple, nous ne présenterons ici que la scène JAVA3D principale.





### 6.2.3.2. Eclairage du soleil

L'éclairage de la scène est créé par deux objets java3d :

- Une lumière ambiante pour voir les parties des objets non éclairées.
- Un "spot light" pour simuler la lumière générée par le soleil

### 6.2.3.3. Orientation des instruments

Les instruments sont simulés de trois façons :

- Par un cône (objet java3D)
- Par une droite (objet java3D)
- Par une simulation de la vue (positionnement de la vue de la scène sur la branche de l'instrument : position + angle de visée + angle de vue)

Il est possible de créer une vue dans une fenêtre séparée. Une nouvelle vue est alors créée et positionnée sur l'attitude de l'instrument.

Le mode visée d'un objet entraîne un placement de l'instrument sur la branche principale de la scène. Lors de l'animation, un algorithme calcule sa position absolue et le positionne.

### 6.2.3.4. Sauvegarde de scénario

La sauvegarde de scénario se fait dans un fichier de style « Properties » qui est ensuite zippé. L'utilisateur choisit son emplacement sur le disque dur de sa machine. Pour la version CDPP, les modèles science comme les cubes et les coupes ont été ajoutés dans la sauvegarde.

### 6.2.3.5. Export des données

Cette fonction permet d'exporter les données orbitales d'un objet de la scène au format ASCII ou VOTable.

L'utilisateur sélectionne un objet, le type de données à exporter et choisit son emplacement sur le disque dur de sa machine.

L'export des données ASCII génère un fichier au format suivant :

3DView CDPP data file

Created on 2015-12-04T14:08:39

Start Time: 2014-06-04T00:00:00

Stop Time: 2014-06-06T00:00:00

Frame: GSE

Center: Earth

Distances unit: km

Body: CLUSTER1

Time X Y Z Vx Vy Vz Lx Ly Lz Dist. Moon Angle:(0.0, 1.0, 0.0)/Sun

2014-06-04T00:00:00 -26853.299 -45976.898 25167.504 0.286 2.717 0.834 -106718.91 29586.431 -59817.71 459443.9864  
89.982644

2014-06-04T00:05:46 -26754.443 -45036.902 25456.07 0.305 2.748 0.816 -106717.553 29595.978 -59799.393 458566.5277  
89.982999

2014-06-04T00:11:32 -26648.996 -44086.047 25738.488 0.324 2.78 0.798 -106716.374 29605.055 -59782.626 457675.9344  
89.983358

### 6.2.3.6. Génération d'images

L'outil génère des images grâce à un canvas3D "offscreen".

Ce canvas est ensuite rattaché à un buffer, ce qui permet de générer des images de différentes qualités.

En haute qualité, l'image générée est trois fois plus grande que le canvas "onscreen". Donc, pour une fenêtre affichée de 800x600, l'image est 2400\*1800 pixels.

On peut noter que cette taille peut dépasser les buffers de la carte vidéo. Pour éviter ce problème, la résolution est limitée comme suit :

Constructeur	Limite
NVIDIA Corporation	4096x4096
ATI Technologies Inc.	2048x2048
Intel	4096x4096
Defaut	1024x1024

L'image générée par le canvas est ensuite enregistrée sur disque dur au format PNG grâce à la fonction apparue depuis le jdk 1.4.2: *ImageIO.write(image,type,fichier)*

### 6.2.3.7. Génération de films

Les films sont générés grâce à deux procédés :

- La sauvegarde d'images
- La génération d'un film à partir d'images

La première partie est identique au chapitre précédent sauf que le type d'image généré est jpeg.

L'API JMF permet ensuite de créer un film quicktime avec un flux d'images au format jpeg.

### 6.2.3.8. Affichages 2D

Pour afficher des variations de données dans le temps (distances, angles), l'API JfreeChart est utilisée.

Elle est chargée à uniquement lors de la demande du graphique 2D et non dès le chargement de l'application 3D.

### 6.2.3.9. Affichage des modèles de analytique intégrés au client 3D

Les modèles suivants sont disponibles :

Corps central	Onde de choc	Magnétopause/pile -up boundary	Autre
Soleil			Spirale de Parker Couche de courant Fronts de CME
Mercure	Kallio & Janhunen Slavin et al. 2009 Moldovan et al 2011.	Kallio & Janhunen Slavin et al. 2009 Moldovan et al 2011.	
Venus	Smirnov et al 1980		
Terre	Sibeck et al. 1991	Shue et al. 1997	Van allen belts, SAA generated from AE8/AP8
Mars	Trotignon et Al. ; Edberg et al 2008	Edberg et al. 2008 ; Dubinin et al. 2006	
Jupiter	Slavin et al. 1985	Slavin et al. 1985	Aurore boréale projetée du 26 Nov. 1998 00:58:08
Saturne	Slavin et al. 1985 Masters et al. 2008	Slavin et al. 1985 Kannani et al. 2010	

#### 6.2.3.9.1. Modèles de chocs/magnétopause/MPB

##### 6.2.3.9.1.1. Modèles de Mercure

###### 6.2.3.9.1.1.1. Choc

Slavin et al., MESSENGER observations of Mercury's magnetosphere during northward IMF, GRL, 36, L02101, doi:10.1029/2008GL036158, 2009. Le choc est modélisé par une conique. Pour la magnétopause le modèle de Howe et Binsack est utilisé. Les paramètres des modèles sont obtenus par un ajustement avec les données de Messenger.

Moldovan et al., Mercury's magnetopause and bow shock from MESSENGER observations, EPSC Abstracts, Vol. 6, EPSC-DPS2011-674, 2011, EPSC-DPS Joint Meeting 2011. Le choc est modélisé par une conique.

#### 6.2.3.9.1.1.2. Magnétopause

Moldovan et al., Mercury's magnetopause and bow shock from MESSENGER observations, EPSC Abstracts, Vol. 6, EPSC-DPS2011-674, 2011, EPSC-DPS Joint Meeting 2011. La magnétopause est donnée par une conique (paraboloïde) et par le modèle de Shue.

#### 6.2.3.9.1.2. Modèle de Venus

##### 6.2.3.9.1.2.1. Choc

L'onde de choc de Venus est issue de Pioneer Venus Orbiter (PVO). Les modèles sont recensés dans un article de Smirnov et al. du JGR volume 85 issue A3 page 7651-7654 de 1980.

Les caractéristiques de la conique sont les suivantes :

- Distance du centre de la planète au point sub-solaire : 8320 km
- Distance au terminateur : 14820 km
- Excentricité : 1.022

Ceci conduit à l'expression suivante (avec les distances en rayon vénusien):

$$Y^2 + (1-E^2)X^2 - 2aEX - a^2 = 0 \text{ ou bien } Y^2 = -(1-E^2)X^2 + 2aEX + a^2$$

avec :  $E = 1.022$  et  $a = -14820 / 6051.8$

#### 6.2.3.9.1.3. Modèles de la terre

##### 6.2.3.9.1.3.1. Onde de choc

Sibeck, D. G., R. E. Lopez, and E. C. Roelof, Solar wind control of the magnetopause shape, location, and motion, J. Geophys. Res., 96, 5489-5495, 1991.

##### 6.2.3.9.1.3.2. Magnétopause de Shue

Shue, J.-H., J. K. Chao, H. C. Fu, C. T. Russell, P. Song, K. K. Khurana, and H. J. Singer, A new functional form to study the solar wind control of the magnetopause size and shape, J. Geophys. Res., in press, 1997.

#### **6.2.3.9.1.4. Modèles de Mars**

##### *6.2.3.9.1.4.1. Choc*

J.G. Trotignon et al. / Planetary and Space Science 54 (2006) 357–369

Edberg et al. , Statistical analysis of the location of the Martian magnetic pileup boundary and bow shock and the influence of crustal magnetic fields, JGR, 113, A08206, doi:10.1029/2008JA013096, 2008. Le choc et la MPB sont modélisés par des coniques en tenant compte l'influence du champ magnétique crustal.

##### *6.2.3.9.1.4.2. Magnetic pile-up boundary*

Edberg et al. , Statistical analysis of the location of the Martian magnetic pileup boundary and bow shock and the influence of crustal magnetic fields, JGR, 113, A08206, doi:10.1029/2008JA013096, 2008.

Dubin et al, Plasma morphology at Mars. Aspera-3 observations, Space Science Reviews, (2006) 126: 209–238, DOI: 10.1007/s11214-006-9039-4. Les frontières des principaux domaines plasma sont déterminées à partir des données ASPERA-3 (MEX). La MPB est modélisée par une conique.

#### **6.2.3.9.1.5. Modèles de Saturne**

##### *6.2.3.9.1.5.1. Choc*

Masters, A.; Achilleos, N.; Dougherty, M. K.; Slavin, J. A.; Hospodarsky, G. B.; Arridge, C. S.; Coates, A. J., An empirical model of Saturn's bow shock: Cassini observations of shock location and shape, Journal of Geophysical Research, Volume 113, Issue A10, CiteID A10210 (JGRA Homepage), 10/2008. Le modèle empirique de Masters et al. utilise une conique pour décrire la géométrie de la surface du choc ainsi que sa position en fonction de la pression du vent solaire. Les paramètres de la conique sont déterminés à partir des données de la sonde CASSINI.

Slavin, J. A., et al. , Solar wind flow about the outer planets: Gasdynamic modelling of the Jupiter and Saturn bow shocks, (1985), J. Geophys.Res., 90(A7), 6275–6286, doi:10.1029/JA090iA07p06275. Slavin et al. proposent des modèles du choc et de la magnétopause de Jupiter et Saturne basés sur la théorie de la dynamique des gaz. Les frontières sont modélisées par des coniques dont la détermination des paramètres tient compte de la pression du vent solaire. Plus précisément, la position des deux frontières varie comme la racine quatrième de la pression dynamique du vent solaire pour Jupiter et la racine sixième pour Saturne.

##### *6.2.3.9.1.5.2. Magnétopause*

Slavin, J. A., et al., Solar wind flow about the outer planets: Gasdynamic modelling of the Jupiter and Saturn bow shocks, (1985), J. Geophys.Res., 90(A7), 6275–6286, doi:10.1029/JA090iA07p06275.

## 6.2.3.9.2. Autres modèles physiques

### 6.2.3.9.2.1. Modèles du soleil

#### 6.2.3.9.2.1.1. Spirale de parker

La spirale est un modèle analytique simple inspiré de la formule 6.98 p225 du Cravens : Parker, E.N., Dynamics of the interplanetary gas and magnetic fields, vol. 128, p 664, Astrophys. J. 1958. Physics of solar system plasmas, Thomas Cravens, Cambridge Atmospheric and Space Science Series, 1997

#### 6.2.3.9.2.1.2. La couche de courant

C'est une représentation 3D de la nappe de courant héliosphérique basé sur la publication THREE-DIMENSIONAL WAVY HELIOSPHERIC CURRENT SHEET DRIFTS, C. Pei, J. W. Bieber, R. A. Burger, and J. Clem. The Astrophysical Journal 744:170 (5pp), 2012 January 10.

#### 6.2.3.9.2.1.3. Les fronts de CMEs

Ce sont des représentations simplifiées de l'observation des CMEs (Coronal Mass Ejection) du vent solaire suivant le modèle fixed point issues d'un catalogue de données HELCATS.

### 6.2.3.9.2.2. Modèles de la terre

Pour la Terre, il est possible d'afficher l'anomalie magnétique sud atlantique (SAA) ainsi que les ceintures de Van Allen. Pour cela, des points ont été générés grâce aux modèles Tsyganenko 96 et AE8/AP8.

Ces points sont enregistrés dans des fichiers qui sont ensuite lus par l'application JAVA. Les fichiers sont situés dans le répertoire web models et se nomment :

- vab\_small\_in\_sm.dat : coquille intérieure de la ceinture intérieure en coordonnées SM
- vab\_small\_out\_sm.dat : coquille extérieure de la ceinture intérieure en coordonnées SM
- vab\_big\_in\_sm.dat : coquille intérieure de la ceinture extérieure en coordonnées SM
- vab\_big\_out\_sm.dat : coquille intérieure de la ceinture extérieure en coordonnées SM
- saa.dat : points de l'anomalie magnétique

### 6.2.3.9.2.3. Format des coquilles des ceintures

Chaque coquille est définie par un ensemble de lignes de force dont les points sont concaténés de la manière suivante :

Nombre de points de la ligne

[Points X, Y, Z val magnétique] \* nb lignes.

#### 6.2.3.9.2.4.Format de la SAA

Hauteur de la couche inférieure (en rayon terrestre). i.e. 1.14  
Hauteur de la couche supérieure (en rayon terrestre). i.e. 1.15  
Nombre de lignes  
[Nombre de points sur la ligne  
[RA DEC] \* nb points  
]\*nb lignes

Les lignes doivent avoir le même nombre de points.

#### 6.2.3.10.Etoiles

Les étoiles sont affichées dans la vie 3D comme PointArray de taille plus ou moins grosses et lumineuse suivant la magnitude visuelle indiquée dans le catalogue.

Ces étoiles sont contenues dans des sous catalogues d'étoiles extraits du "Master catalog" tycho2 de la NAIF via l'outil SPICE starsb.

Le catalogue tycho2 est basée sur les données de la mission ESA Hipparcos et est disponible à l'adresse suivante: [ftp://naif.jpl.nasa.gov/pub/naif/generic\\_kernels/stars/](ftp://naif.jpl.nasa.gov/pub/naif/generic_kernels/stars/)

Le fichier contenant la liste des sous-catalogues est dans le répertoire des fichiers d'orbites *orbit\_files* et se nomme *starfiles.lst*.

Les catalogues *all\_vm* désignent les catalogues contenant toutes les étoiles de magnitude visuelle inférieure au nombre indiqué.

Les catalogues *corot\_vm* contiennent les étoiles comprises entre les coordonnées  $-15^{\circ}$  à  $15^{\circ}$  de latitude et de  $89^{\circ}$  à  $119^{\circ}$  et de  $269^{\circ}$  à  $299^{\circ}$  EST de longitude. On peut voir que pour afficher les zones observables par COROT avec les étoiles de magnitude inférieure à 10, on utilise deux sous catalogues : un pour les longitudes entre 89 et 119 et un pour celles entre 269 et 299.

Voici un exemple de fichier de configuration utilisé pour générer les sous-catalogues :

```
\begindata

  STARSB_MASTER_FILE = 'tycho2.bin'
  STARSB_OUTPUT_FILE = 'tycho2_corot_vm10_104.dbk'
  STARSB_ETCAL_TIME   = '2000-JAN-1 12:00:00'
  STARSB_WESTRA       = 89
  STARSB_EASTRA       = 119
  STARSB_STHDEC       = -15.0
  STARSB_NTHDEC       = 15.0
  STARSB_VMMAX        = 10.0

\beginntext
```

### 6.2.3.11. Génération de timetables

Grâce à une interface, 3DView rend facile la génération de timetables. Il suffit de positionner l'animation au temps début ou fin voulu et de cliquer sur un bouton "Add time" pour générer les temps début et fin successivement. Ensuite, un bouton "Generate timetable" permet de sauvegarder le fichier sur le disque dur.

### 6.2.3.12. Ajout d'objets inconnus de 3DView à partir d'un fichier utilisateur

L'utilisateur a la possibilité d'ajouter un spacecraft à la scène en indiquant un nom, et un fichier d'orbite accessible depuis sa machine. Le format peut être texte (temps, x, y, z) ou VOTable. Le fichier doit être dans le même repère que la scène.

### 6.2.3.13. Ajout de l'attitude d'un objet de 3DView à partir d'un fichier utilisateur

L'utilisateur a la possibilité d'ajouter à un spacecraft de la scène une attitude sélectionnant l'objet concerné et en indiquant le format et le contenu du fichier.

Le format peut être VOTable ou ASCII. Il peut contenir des matrices (Time,xx,xy,xz,yx,yy,yz,zx,zy,zz) ou des quaternions (Time,x,y,z,w).

### 6.2.3.14. Ajout d'une ligne de champ à partir d'un fichier utilisateur

Cette fonction a été réalisée pour intégrer des résultats d'un générateur de lignes de champs dans l'héliosphère. Chaque fichier contient une seule ligne de champ avec sur chacune la position du point, la valeur du champ et éventuellement d'autres valeurs. Le format est le suivant :

R(R\_SUN) Theta(rad) Phi(rad) Br(G) Btheta(G) Bphi(G)

Si d'autres valeurs suivent, l'utilisateur doit choisir la valeur qu'il veut utiliser pour la représentation couleur.

### 6.2.3.15. Ajout de champs de vue d'un instrument à partir d'un fichier utilisateur

Pour ajouter des instruments sans avoir à saisir les caractéristiques une par une, l'utilisateur peut uploader un fichier contenant les définitions de plusieurs instruments à partir d'un fichier.

Le fichier doit contenir des lignes au format suivant : Vx,Vy,Vz, angle plein, commentaire.

Exemple de ligne: 1 ;0 ;0 ;90 ;X axis 90°

### 6.2.3.16. Décalage des objets dans le temps

Avant l'ajout d'un objet (modèles, valeurs physiques d'observation, satellites), il est possible de préciser un temps et un pas précis pour chaque mission. Pour cela, toutes les interfaces d'ajout et de filtre de temps prennent ces paramètres en compte.

Par exemple, pour l'ajout d'un satellite, un filtre de temps empêche de sélectionner un satellite qui n'aurait pas de données dans l'intervalle de temps de référence. Un bouton sur chaque ligne de sélection permet de préciser ces paramètres. L'objet ajouté est alors automatiquement décalé en temps dans la scène.

Les objets décalés en temps portent un signe distinctif qui permet de les différencier des autres objets du même type. Les trajectoires des satellites donc les données sont futures à la scène sont vertes tandis que celles passées sont rouges.

### 6.2.3.17. Ajout de satellites de la terre à partir d'un fichier utilisateur au format TLE

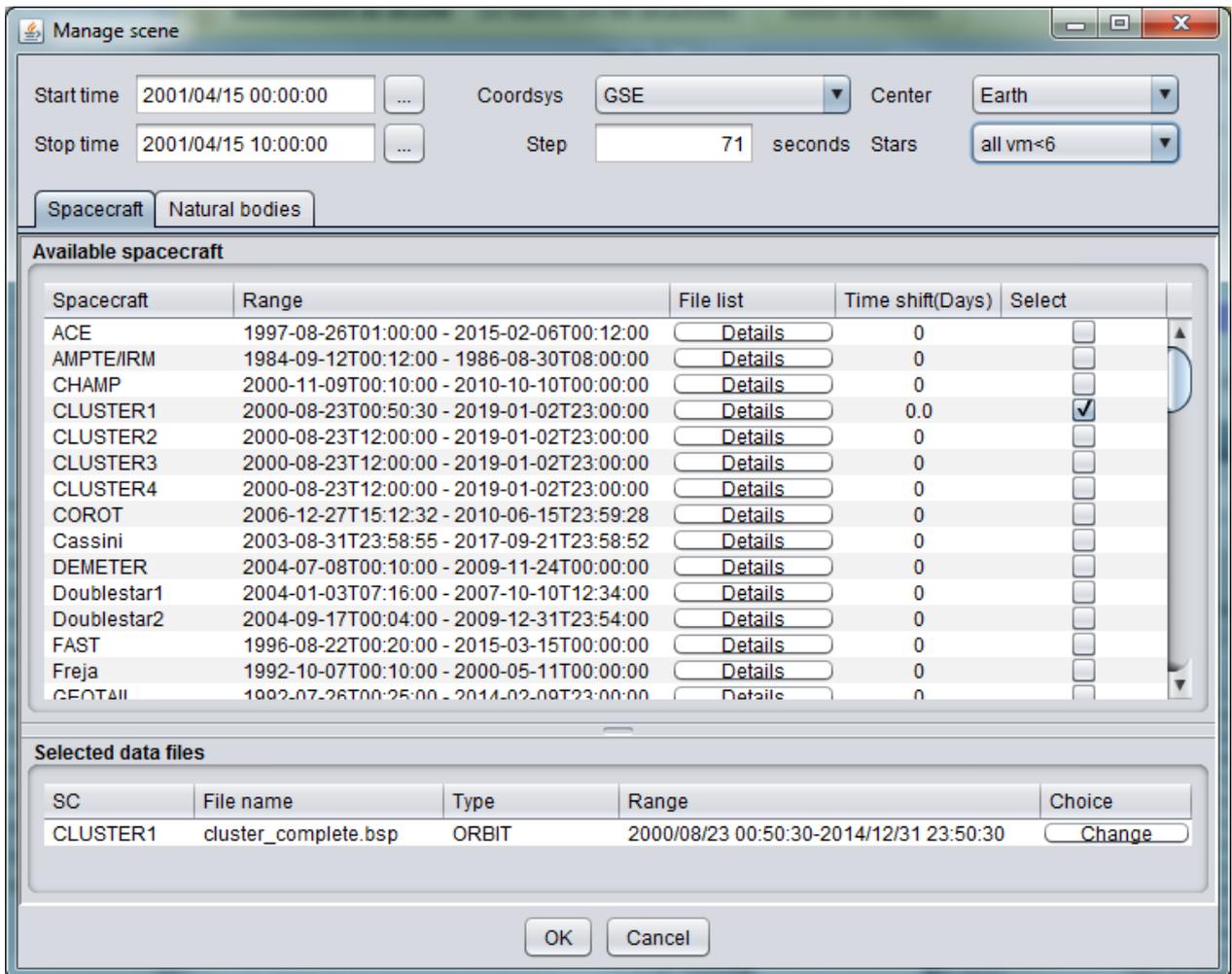
L'utilisateur a la possibilité d'ajouter un spacecraft à la scène en indiquant un nom et un fichier d'orbite accessible depuis sa machine. Le fichier doit être en ASCII au format TLE.

Voici un exemple de fichier au format TLE:

```
NOAA 14
1 23455U 94089A 97320.90946019 .00000140 00000-0 10191-3 0 2621
2 23455 99.0090 272.6745 0008546 223.1686 136.8816 14.11711747148495
```

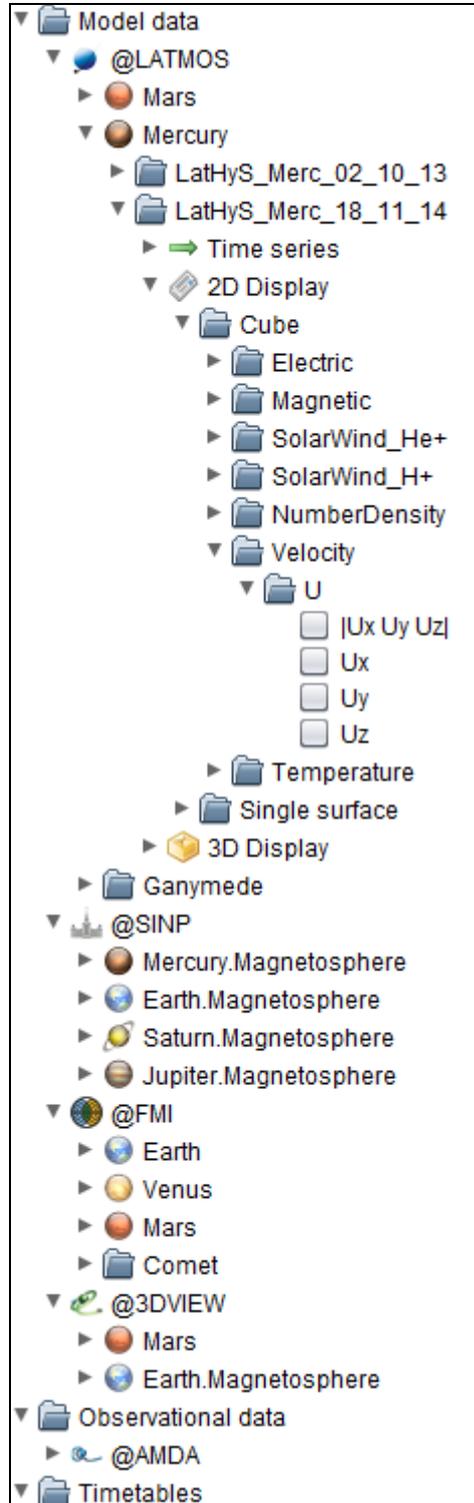
## 6.3.LES WEB SERVICES 3DVIEW

Le client 3D liste et récupère les données concernant la mécanique spatiale via l'IHM suivante :



Cette IHM et les traitements qui en découlent comme l'ajout de sonde, l'affichage de planètes, des étoiles, ... ne pourrait exister sans le serveur 3DView qui fournit ces informations via le web service d'orbitographie.

Les données physiques sont, elles, accédées via un arbre de données :



### 6.3.1. Description

L'interface du web service est définie par un fichier WSDL. Pour le 3DView, il est accompagné de deux fichiers xsd (cdpp3dview\_types.xsd et cdpp3dview\_params.xsd) de définition des structures de données et d'un API d'implémentation à utiliser par le client.

Les services sont définis dans le document DR3.

### 6.3.2. Implémentation client

Pour se connecter au serveur le client intègre l'API client du web service. Il l'initialise avec l'adresse du serveur transmise dans le lancement JNLP et la connexion est établie.

### 6.3.3. Implémentation serveur

Le serveur fournit les services grâce à une application web hébergée sous Apache Tomcat.

Pour répondre aux requêtes clientes, le serveur obtient les données dans des fichiers locaux qui peuvent être des SPICE kernels ou des fichiers de configuration. Pour accéder aux SPICE kernels, des bibliothèques orbito en FORTRAN ont été utilisées. Ce sont des programmes natifs qui sont exécutés pour lire les noyaux SPICE. Un chargement en mémoire était impossible car chaque demande du client étant unique et le contexte de l'application web étant partagée entre tous, il y aurait eu des problèmes lors d'accès concurrents. En effet, l'API SPICE conserve une liste globale des noyaux chargés en mémoire.

Pour les données contenues dans des fichiers de configuration, une lecture directe depuis l'application web est réalisée.

Voici la répartition des types d'accès en fonction des demandes :

	Fichiers de configuration	Exécution de programme natif
Orbite		X
Attitude		X
Etoiles		X
Recherche de comète/astéroïde		X
Liste des sondes	X	
Taille des planètes		X
Orientation des panneaux solaires		X
Liste des instruments	X	
Liste des kernels étoiles	X	
Liste des planètes	X	
Liste de frames	X	

Voici une description des fichiers de configuration qui permettent de lister et personnaliser les objets affichés. Ces fichiers sont localisés à trois endroits différents :

- Dans l'appli web WEB-INF/classes :
  - A.1.*bodies.properties* : liste et caractéristiques des sondes, planètes à afficher...
  - A.2.*services.properties* : noms et paramètres pour lancer les programmes natifs
  - A.3.*conf.properties* : paramètres de l'application web
- Dans le répertoire \$HOME/i3dv\_files/conf
  - A.4.*updfilesscweb.conf* : liste des données des missions sscweb à mettre à jour.
  - A.5.*updfiles.conf* : liste des autres missions à mettre à jour.
- Dans le répertoire \$HOME/i3dv/server/web/conf
  - A.6.*groundstations.conf*: liste des stations terrestres et leurs propriétés
  - A.7.*scparams.conf*: propriétés spécifiques aux sondes
  - A.8.*starfiles.conf*: liste des noyaux d'étoiles
  - A.9.*scinstruments.conf* : liste les instruments des missions et leur représentation
  - A.10.*groundlabels.conf* : liste des labels au sol par mission
  - A.11.*frames.conf* : liste des différents systèmes de coordonnées
  - A.12.*projmaps.conf* : liste des cartes de projection
  - A.13.*crmmaps.conf* : liste des types de cartes de Carrington

### 6.3.3.1.Fichiers de l'application web

#### 6.3.3.1.1.bodies.properties

Voici un exemple de contenu de ce fichier :

```
#available bodies in services, do not add space between commas
m3dv.bodies.art=-1,-8,-21,-31,-32,-41,-55,-77,-82,-92,-183,-185,-194,-196,-226,\
-234,-235,-236,-337,-338,-339,-340,-341,-248,\
-551,-486,-489,-552,-553,-554,-555,-78,-556,-74,-76,-61,-907,-202,-68,-69,-999,-94,-20180731,\
-110422,-110423,-111004,-128140,-128382,-106893,-13,\
-128405,-128368,-124285,-122161,-126113,-115200,-125635,-128810,-136395,-139452,-139451,-139453,-126998,-125280,\
-301,-302,-140482,-140483,-140484,-140485
m3dv.bodies.nat=10,199,299,399,301,499,401,402,599,501,502,503,504,505,514,699,601,602,603,604,605,606,607,608,799,
701,702,703,704,705,899,801,802,999,901,1000012,2000021,2002867,1000036,1000034
# spacecraft names */
m3dv.bodies.art.name=GEOTAIL,WIND,SOHO,Voyager_1,Voyager_2,MEX,ULYSSES,Galileo,Cassini,ACE,CLUSTER1,CLUSTER2,C
LUSTER3,CLUSTER4,Rosetta,\
Stereo-A,Stereo-B,MESSENGER,THEMIS-A,THEMIS-B,THEMIS-C,THEMIS-D,THEMIS-E,\
VEX,COROT,Herschel,Planck,SVOM,Jason                                2,INTERBALL-Auroral,INTERBALL-
Tail,Giotto,PICARD,MRO,MSL,Juno,JUICE,MAVEN,MMO,MPO,Solar Orbiter,MGS,Solar Probe Plus,\
ISEE-1,ISEE-2,ISEE-3,Doublestar1,Doublestar2,IMP-8,POLAR,\
CHAMP,DEMETER,FAST,Freja,IMAGE,AMPTE/IRM,Oersted,Reimei,SDO,SwarmA,SwarmB,SwarmC,TimeD,TRACE,\
```



### 6.3.3.1.2.services.properties

Ce fichier est utilisé pour pouvoir appeler les programmes natifs d'accès aux noyaux SPICE. Il contient les noms des paramètres à passer, les répertoires des exécutables, librairies, ...

Voici un exemple des valeurs pour les paramètres. Les noms sont suffisamment explicites pour comprendre leur fonction :

```
m3dv.native.libs.dir.path=LD_LIBRARY_PATH=/usr/lib
m3dv.native.exes.dir.path=/home/i3dv_dev/i3dv/server/web/native/exe/bin
m3dv.native.exe.orb=listOrbData.exe
m3dv.native.exe.orb2=listOrbData2.exe
m3dv.native.exe.att=listAttData.exe
m3dv.native.exe.att2=listAttData2.exe
m3dv.native.exe.eme=listEmeAttData.exe
m3dv.native.exe.stars=listStars.exe
m3dv.native.exe.size=getSize.exe
m3dv.native.exe.insts=getFov.exe
m3dv.native.exe.inregion=listIdsInRegion.exe
m3dv.native.exe.newFrameAtt=convertFrameData.exe
m3dv.native.exe.newFrameOrb=convertFrameData.exe
m3dv.native.orb.param=I3DV_ORBIT_DIR
m3dv.native.log.param=I3DV_LOG_FILE
m3dv.native.log.path=/home/i3dv_exp/i3dv_files/logs/i3dvdev_web_native.log
m3dv.model.native.exe.tsy=tsy96.exe
m3dv.model.native.exe.cain=cain_impex.exe
m3dv.model.native.exe.cain.coeff.path=/home/i3dv_dev/i3dv/server/web/native/exe/models/FSU_mars90
```

### 6.3.3.1.3.conf.properties

Ce fichier contient essentiellement la localisation des différents fichiers de configuration :

```
m3dv.conf.dir.path=/home/i3dv_exp/i3dv/server/web/conf
m3dv.conf.scparams.file.path=/home/i3dv_exp/i3dv/server/web/conf/scparams.conf
m3dv.conf.frames.file.path=/home/i3dv_exp/i3dv/server/web/conf/frames.conf
m3dv.conf.scinsts.file.path=/home/i3dv_exp/i3dv/server/web/conf/scinstruments.conf
m3dv.conf.groundstations.file.path=/home/i3dv_exp/i3dv/server/web/conf/groundstations.conf
m3dv.conf.projmaps.file.path=/usr/share/tomcat6/webapps/ESSAI/images/textures/projmaps.conf
m3dv.conf.crmaps.file.path=/home/i3dv_dev/i3dv/server/web/conf/crmaps.conf
m3dv.conf.stars.file.path=/home/i3dv_exp/i3dv/server/web/conf/starfiles.conf
m3dv.conf.web.pub.dir.path=/usr/share/tomcat6/webapps/ESSAI/pub
m3dv.conf.rep.dir.path=/home/i3dv_exp/i3dv_files/commonres
m3dv.orbito.dir.path=/home/i3dv_exp/i3dv_files/orbitfiles
m3dv.lst.naifidx.file.path=/home/i3dv_exp/i3dv_files/orbitfiles/naiffiles.lst
m3dv.crmaps.dir.path=/home/i3dv_exp/i3dv_files/carringtonmap
m3dv.cmefits.dir.path=/home/i3dv_exp/i3dv_files/fitcatalog/CME
m3dv.lst.coverageidx.file.path=/home/i3dv_exp/i3dv_files/orbitfiles/sccoverage.lst
m3dv.lst.genkernels.file.path=/home/i3dv_exp/i3dv_files/orbitfiles/spice/kernels.txt
m3dv.test=false
```

## 6.3.3.2. Configuration des données

### 6.3.3.2.1. groundstations.conf

Il liste les stations au sol de la terre.

Pour chaque station, on peut paramétrer le nom, la longitude, la latitude, l'angle de visibilité et la couleur du cône qui sera affiché.

#### Extrait :

```
#format is: scid, name, RA, DEC, visibility angle (deg) ,color
-226;Kiruna;20.42;67.83;60;#0000FF;1000
-226;Alcantara;-43.5;-2.5;60;#00FF00;1000
-551;Aussaguel;1.5;43.5;60;#FF0000;1000
```

### 6.3.3.2.2. scparams.conf

Il permet de personnaliser l'interface en fonction de la sélection des missions. Par exemple, lorsqu'une mission est sélectionnée, le système de coordonnées, les corps centraux proposés et les planètes sélectionnables seront présélectionnés et filtrés.

#### Extrait :

```
#List specific parameters for each spacecrafts
#Format:
#   naifId ; coordsys ; center ; planetId[ planetId[ planetId[ ...]];[specific star choice index only for Corot for the moment.]
#
# coordsys: EMEJ2000 1, ECLIPJ2000 2, GSE 3, MSO 4, VSO 5, GSM 6, SM 7
# center: SUN 10, MERCURY 199, VENUS 299, EARTH 399, MARS 499, 599, 699, 799, 899, 999, ...
#
#ROSETTA
-226;2;10;299 399 499 599 1000012 2000021 2002867
#COROT
-551;1;399;301;10
#MEX
-41;4;499;399 401 402
```

### 6.3.3.2.3. starfiles.conf

Il liste les noyaux d'étoiles disponibles qui seront proposée via l'interface web à l'utilisateur.

En séparant les noyaux par ":", on peut en spécifier plusieurs pour un même choix.

Un 3<sup>ème</sup> paramètre dans la ligne spécifie le noyau à choisir par défaut sauf paramétrage particulier pour une mission dans scparams.conf.

**Extrait :**

```
#spice dbk star files
#put a ;<anything> to tag default selection

all vm<5;tycho2_vm5.dbk
all vm<6;tycho2_vm6.dbk;S
all vm<7;tycho2_vm7.dbk
corot vm<7;tycho2_corot_vm7_104.dbk:tycho2_corot_vm7_284.dbk
corot vm<8;tycho2_corot_vm8_104.dbk:tycho2_corot_vm8_284.dbk
```

**6.3.3.2.4.scinstruments.conf**

Il définit les instruments des sondes permettant de préremplir la boîte de dialogue View instrument direction.

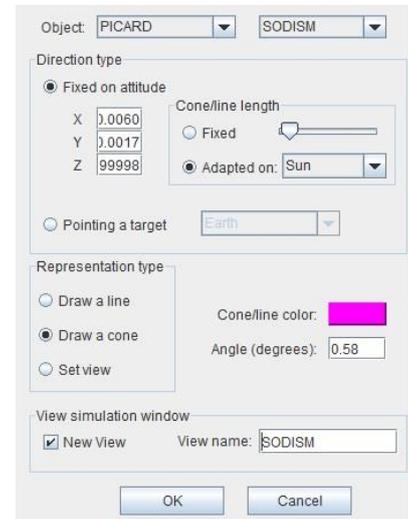
Chaque ligne contenue peut avoir deux formats différents.

Soit descriptif :

idnaif ; nom instrument ; CONE|LINE ; couleur ; x ; y ; z ; HFov(Deg) ; VFov(Deg); FrontClip(km); BackClip(km) ; type de longueur du cône (Y/N) ; id naif d'adaptation de longueur ou longer fixe de 0 à 100.

**Exemple :**

```
-556;SODISM;CONE;#FF00FF;-0.006;-0.0017;-
0.99998;0.58;0.58;10;150000000>true;10
```



Le backclip est à 150 millions de km, la distance terre soleil pour que l'instrument puisse voir le soleil depuis la terre. Normalement, le rapport Frontclip/Backclip doit être de 3000 au plus.

Soit référentiel.

Chaque ligne référence uniquement un instrument défini dans des spice kernels(fk,ik) préchargés dans le fichier \$HOME/i3dv\_files/orbitfiles/spice/kernels.txt.

Le format est le suivant :

SC naifld ; instrument naifld; instrument name ; representation(CONE, LINE) ; color ; FrontClip(km); BackClip(km) ; Length adaptative(Y/N) ; naif id of adaptation or length(0/100)

**6.3.3.2.5.groundlabels.conf**

Il liste les labels sur la planète centrale pour certaines sondes.

Pour chaque label, on peut paramétrer le nom, la longitude, la latitude et la couleur du texte qui sera affiché.

**Extrait :**

```
-74;Viking 1;312.05;22.46;#FFFFFF
-74;Viking 2;-225.99;48.269;#FFFFFF
```

**6.3.3.2.6.frames.conf**

Il liste les systèmes de coordonnées disponibles pour les services listOrbData2, listAttData2, listFrames2, listNewFrameAtt, listNewFrameOrb.

Ces systèmes sont soit définis en natif dans la spicelib (J2000, ECLIPJ2000) soit déclarés dans le fichier impex.tf situé dans le répertoire \$HOME/i3dv\_files/orbitfiles/spice et inclus dans le fichier \$HOME/i3dv\_files/orbitfiles/spice/kernels.txt.

Pour chaque système, on doit paramétrer l'id, le nom, le centre et la description.

**Extrait :**

```
#Format:
#  naifld ; name ; center naifld; description
#
# SUN
1;J2000;10;Earth mean equator, dynamical equinox of J2000
17;ECLIPJ2000;10;Ecliptic coordinates based upon the J2000 frame
1600010;HEE;10;Heliocentric Earth Ecliptic
1601010;HEEQ;10;Heliocentric Earth Equatorial
1602010;HCI;10;Heliocentric Inertial
#MERCURY
1600199;MEME;199;EME2000 centered on Mercury
1601199;MECLIP;199;ECLIPJ2000 centered on Mercury
1603199;MESO;199;Mercury-centric Solar Orbital
#VENUS
1600299;VSO;299;Venus-centric Solar Orbital
1601299;VME;299;Venus Mean Equator
#EARTH
1600399;GSE;399;Geocentric Solar Ecliptic
1601399;EME;399;Earth Mean Equator and Equinox
```

**6.3.3.2.7.crmaps.conf**

Il liste les types de cartes de Carrington disponibles par le service listCRMapTypes et visible sur le client via le menu Science/load Carrington map.

Chaque type carte est identifié par l'Id du corps central, le nom de l'observateur, l'instrument de mesure, la longueur d'onde et le path du catalogue permettant de lister les cartes de Carrington d'un même type.

**Extrait :**

```
#list available Carrington map types
#format is: observatory, instrument, measurement and data path

10;GONG;MAG;Mag;GONG/MAG
10;SOHO;EIT;195;SOHO/195A
10;SOHO;EIT;304;SOHO/304A
10;SOHO;MDI;Mag;SOHO/MDI
10;SDO;HMI;Mag;SDO/MAG
10;STEREO-A;EUVI;195;STEREO-A/195A
10;STEREO-A;EUVI;304;STEREO-A/304A
10;STEREO-B;EUVI;195;STEREO-B/195A
10;STEREO-B;EUVI;304;STEREO-B/304A
```

**6.3.3.2.8.projmaps.conf**

Il liste les maps disponible par le service listProjMaps et visible sur le client via le menu Science/load map.

Pour chaque carte, on peut paramétrer le corps, le nom, l'altitude, la description puis le nom du fichier de texture (jpg, gif ou png) situé dans l'appliweb (images/textures).

**Extrait :**

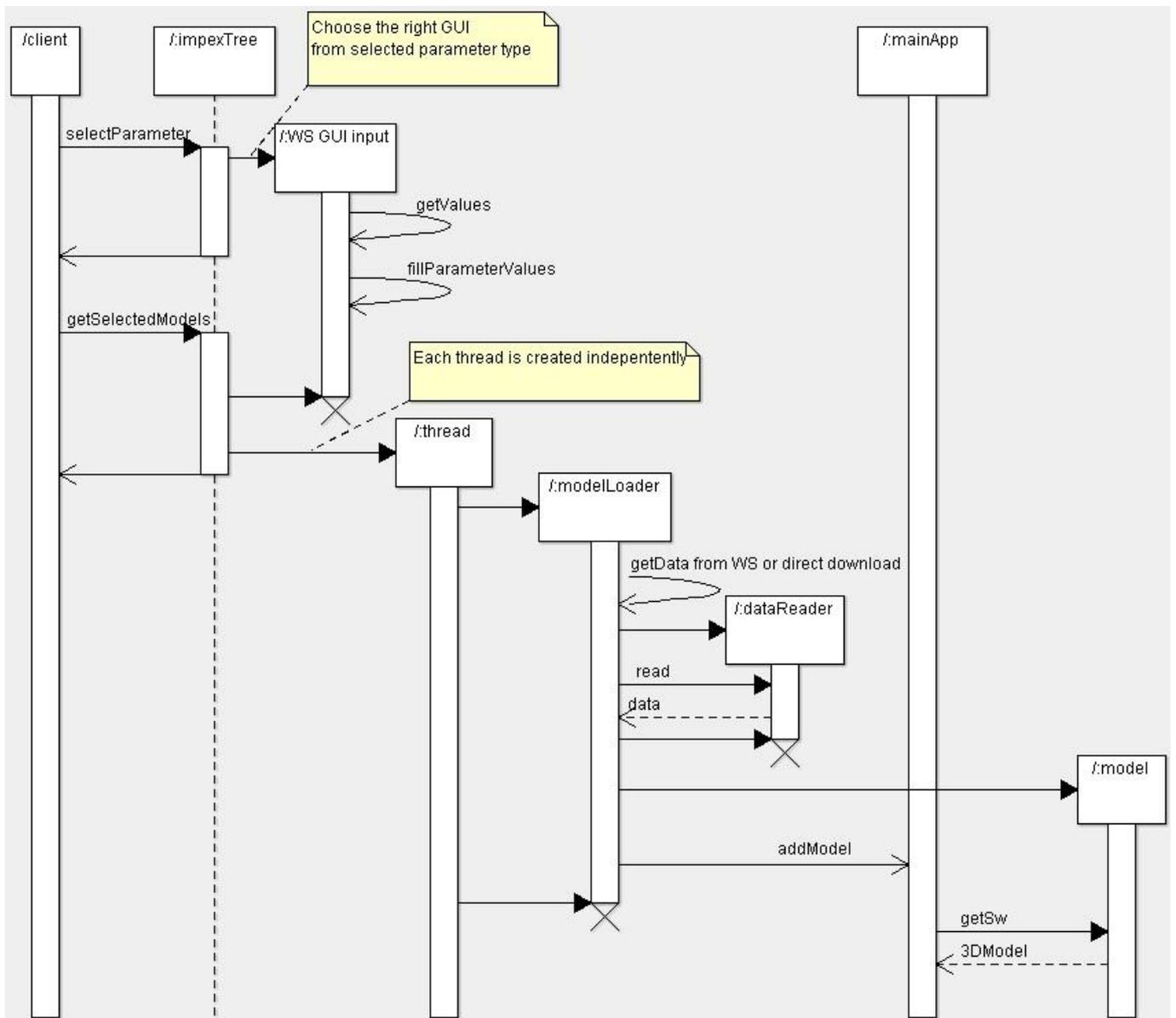
```
#list projection maps and their properties
#format is: body naif id, name, altitude from surface (km), description, path from texture directory

10;Default;0;Standard colored texture;10_texture.jpg
199;Default;0;Standard colored texture;199_texture.jpg
299;Default;0;Standard colored texture;299_texture.jpg
399;Default;0;Standard colored texture;399_texture.jpg
399;Magnetic;0;Custal magnetic anomalies;399_Magnetic_texture.jpg
399;Height;0;Height map;399_Custom_texture.jpg
399;Night;0;Night city map 1024x512;399_earthlights1k.jpg
399;Clouds;300;Cloud map 1024x512;399_earthcloudmap.png
499;Default;0;Standard colored texture;499_texture.jpg
499;Crustal Morschhauser;180;Morschhauser map at 180km;499_morschhauser_180.jpg
499;Crustal Connerney;400;Connerney J.E.P. et al.(2005) Proc. Nati. Acad. Sci USA, 102, No. 42, 14970-14975.;499_Magnetic_texture.jpg
```

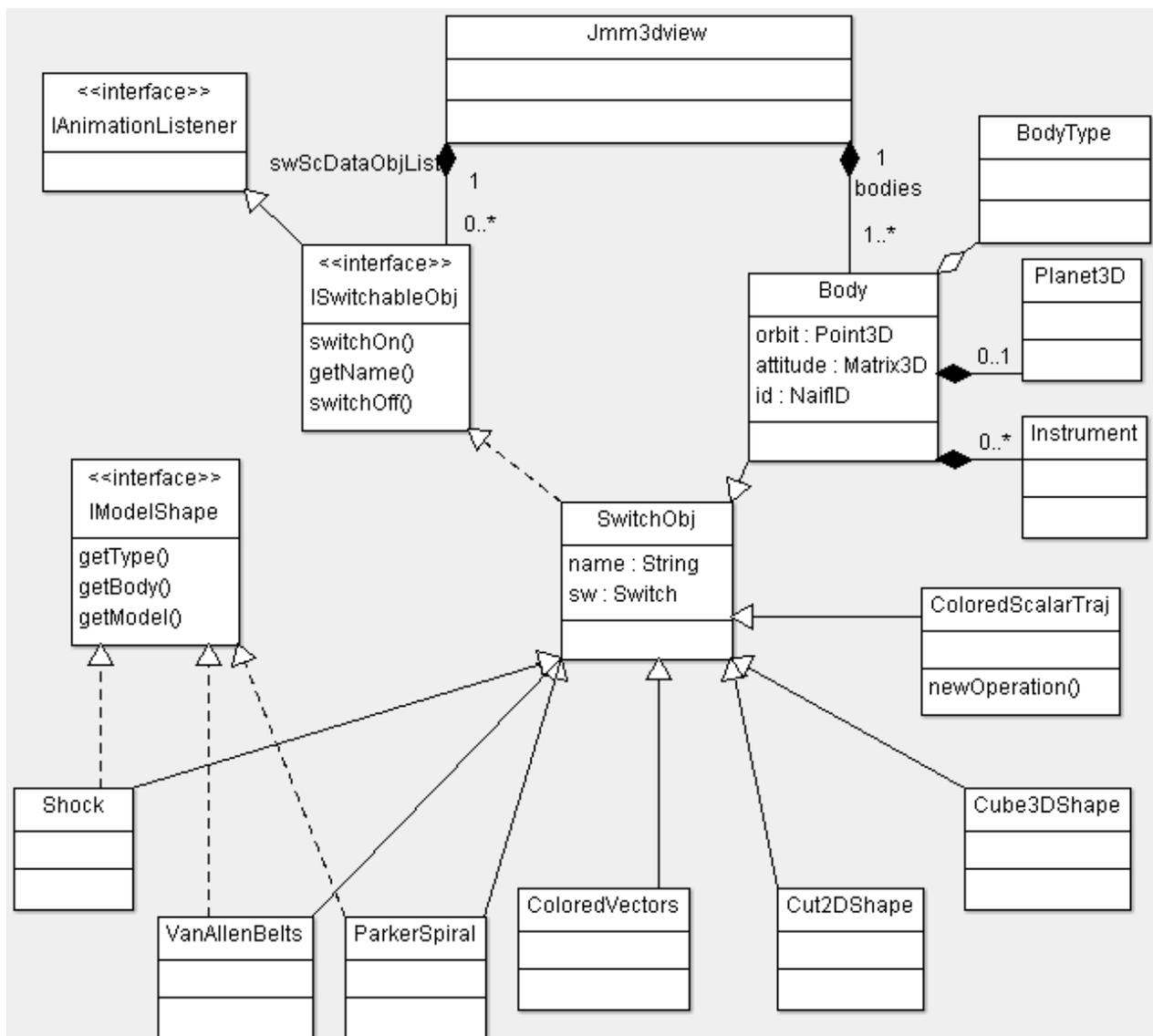
### 6.4.AFFICHAGE DES DONNEES AU FORMAT IMPEX

Pour afficher les données de simulation (LATMOS, FMI, SINP), et d'observation (AMDA, CLWeb, CDAWeb), il faut passer par une phase de sélection des données dans un arbre, une phase de récupération des données par des web services ou en téléchargement direct puis une phase de construction de la représentation graphique. Un mode spécifique permet aussi d'afficher des données d'AMDA grâce à une alimentation par Hub SAMP.

Voici un diagramme de séquence décrivant les différents processus permettant de sélectionner, récupérer et afficher les modèles des différents fournisseurs IMPEX :



Nous pouvons observer la structure de classe basique des différents objets 3D affichés dans ce diagramme de classe :



On peut y voir les principaux objets affichés :

- Body : planètes, spacecrafts, comètes, ...
- Des modèles avec structure 3D calculée : les chocs, ceintures et spirale de parker
- Des données science : Cut2DShape, Cube3DShape, ColoredVectors, ...

Tous ces objets héritent de l'interface ISwitchableObj qui permet d'afficher ou non un objet et tous ceux dépendant du temps implémentent les méthodes de IAnimationListener.

## 6.4.1. Les types de données

Il y a deux principaux types de données :

- les données d'observation venant d'AMDA, de CLWeb, du CDAWeb ou d'un fichier utilisateur
- les données de simulation venant du LATMOS, du FMI ou du SINP.

### 6.4.1.1. Les données d'observation

Le client 3D peut se connecter à quatre services pour récupérer des données d'observation : AMDA, CLWeb, CDAWeb et CSA. Les données disponibles sont affichées dans l'arbre de sélection.

L'utilisateur peut aussi ajouter des données sélectionnant un fichier sur son disque dur.

#### 6.4.1.1.1. AMDA, CLWeb

Les services AMDA et CLWeb sont semblables. Ils sont accédés par le protocole SOAP et permettent de récupérer la liste des paramètres affichables par la méthode getAMDATree. Les données sont ensuite obtenues avec getParameter. L'adresse des web services AMDA est la suivante : [http://cdpp1.cesr.fr/AMDA-NG/public/wsd/Methods\\_AMDA.wsdl](http://cdpp1.cesr.fr/AMDA-NG/public/wsd/Methods_AMDA.wsdl). L'adresse des services CLWeb est : [http://clweb.cesr.fr/Methods\\_CLWEB.wsdl](http://clweb.cesr.fr/Methods_CLWEB.wsdl)

Les données récupérées sont des liens vers un ou plusieurs fichiers NetCdf couvrant la période de temps demandée.

#### 6.4.1.1.2. CDAWeb

Cette base de données d'observation est accédée avec SOAP en JAX-RPC. Son adresse est la suivante : <http://cdaweb.gsfc.nasa.gov/WS/jaxrpc?WSDL>.

#### 6.4.1.1.3. Cluster Science Archive

Les données du centre cluster contiennent beaucoup de données et sont accessible avec des requêtes http. La description du service est disponible sur [https://csa.esac.esa.int/csa/aio/html/home\\_main.shtml](https://csa.esac.esa.int/csa/aio/html/home_main.shtml)

#### 6.4.1.1.4. Par fichier

Le chargement par fichier permet de charger des données avec le temps en première colonne au format ISO. Les données sont ensuite affichées sous forme de scalaire : ligne colorée perpendiculaire à l'orbite.

### 6.4.1.2. Les données de simulation

Il y a deux traitements différents suivant que les données sont affichables directement (precomputed) ou doivent faire l'objet d'un appel à un web service (on the fly).

#### 6.4.1.2.1. Les données precomputed

Il y a quatre types de données pré-calculées : les cubes, les coupes 2D, les timeseries et les spectrogrammes.

Toutes ces données sont affichables directement sans interface intermédiaire.

Les cubes et les coupes sont accédés directement grâce à l'URL des granules du tree.xml tandis que les Timeseries et les spectra sont récupérés via la méthode des webservices fournisseur getFileUrl.

#### 6.4.1.2.2. Les données on the fly

Les données on the fly se basent sur des simulations déjà effectuées (Run) ou assimilés comme tels (Calcul à la volée du SINP ou du LATMOS). Lorsque l'utilisateur sélectionne un cube, une interface lui est proposée, lui permettant de choisir le service à appeler et les paramètres à envoyer. Ces services et leurs paramètres diffèrent entre les fournisseurs. Ils sont décrits dans le chapitre suivant.

Pour les données nécessitant en entrée un ensemble de points dans un fichier VOTable, un appel au web service 3DView getOrbUrl est effectué.

### 6.4.2. L'arbre de sélection

L'arbre de sélection permet d'afficher les données disponibles des différents fournisseurs (LATMOS, FMI, SINP, AMDA, CLWeb et 3DView) via le fichier tree.xml ou les web services.

Les fichiers tree.xml sont mis en cache dans l'espace prévu à cet effet. Si les tree.xml ne sont pas accessibles via le web, les versions du cache seront utilisées.

Les valeurs précalculées sont affichables directement. Il est possible de sélectionner plusieurs paramètres et de tous les ajouter en même temps à la scène.

#### 6.4.2.1. Les nœuds génériques

Les nœuds ayant des fils constituent les « branches » de l'arbre. Ils peuvent soit être de simples objets textes, soit des objets plus complexes. Ils ne sont pas sélectionnables.

Les feuilles de l'arbre sont des CheckBox dont la sélection est détectée par un Listener. Chaque feuille représente un paramètre qui peut être un cube, un spectre, une série temporelle, un modèle, ...

## 6.4.2.2. La construction de l'arbre

### 6.4.2.2.1. Les nœuds venant des tree.xml

L'affichage de l'arbre des données du FMI, LATMOS et SINP implique la lecture d'un fichier tree.xml. Ce fichier est traité par un parser SAX commun.

Deux infobulles sont affichées, une sur le nœud simulation et une sur la feuille. Les informations affichées pour la simulation sont les mêmes que celles affichées dans AMDA. Celles affichées au niveau feuilles contiendront l'unité, le type, la période temporelle si elle existe et éventuellement d'autres champs à déterminer.

Les feuilles sont issues des parameterKey (NumericalOutput/Parameter/ParameterKey).

Si un parameterKey est détecté comme étant un vecteur, deux feuilles virtuelles sont créées, permettant de sélectionner les trois composantes pour afficher le module ou les vecteurs, et les trois paramètres sélectionnables individuellement.

### 6.4.2.2.2. Les nœuds venant du CDAWeb

L'arbre est construit en faisant appel successifs à des méthodes du web service : getMissionGroupDescriptions(), getSources(), getInstrumentTypes, etc.

La méthode VariableDescription.getShortDescription() permet de différencier un vecteur d'un scalaire (s'il contient « \_vec »).

### 6.4.2.2.3. Les nœuds venant du CSA

Les accès utilisent le mode synchrone dont on peut obtenir la description à l'adresse <https://csa.esac.esa.int/csa/aio/html/datarequests.shtml>.

L'accès est assez lent car les requêtes sont très basiques. Il n'est pas possible d'obtenir les paramètres d'un satellite mais il faut d'abord lister les datasets disponibles, puis faire une requête par dataset pour en obtenir les paramètres.

Ne sont retenus que les datasets de category Science et general ainsi qu'un time\_resolution > 0. Pour les paramètres, seuls ceux du type Data sont sélectionnés.

### 6.4.2.2.4. Les nœuds venant d'Amda et CLWeb

La liste des paramètres affichables est récupérée par la méthode commune getParameterList de chaque service. On peut y récupérer des scalaires, des vecteurs, des spectrogrammes et des timetables. Si un login Mot de passe est entré dans les paramètres généraux, les paramètres et timetable d'un utilisateur peuvent aussi être affichés.

#### **6.4.2.2.5. Les nœuds venant des web services LATMOS, FMI, SINP et 3DView**

Pour pouvoir faire appel aux web services générant des données "on the fly", des feuilles virtuelles sont créées permettant de faire appel aux méthodes associées.

Lorsqu'une CheckBox est cochée, une méthode du paramètre permet de récupérer l'interface correspondante. Cette interface, affichée à côté de l'arbre, contient des paramètres qui serviront d'entrée lors de l'appel des web services pour récupérer les données.

#### **6.4.2.3. Les filtres**

Les filtres permettent de n'afficher dans l'arbre que les données compatibles avec certains critères. Les trois principaux filtres sont le filtre temporel, le filtre par région et le filtre d'objet. Tous les filtres sont débrayables par l'utilisateur.

Le filtre temporel n'affiche que les données couvrant la période de la scène 3D affichée.

Le filtre par région n'affiche que les simulations correspondant à l'objet central de la scène 3D.

Le filtre par objet n'affiche que les données des satellites affichés dans la scène.

L'activation ou la désactivation d'un filtre implique la régénération totale de l'arbre.

#### **6.4.2.4. Les interfaces d'entrée des Web services**

Les interfaces sont au nombre de 3. Chacune gèrent les entrées de plusieurs web services de différents fournisseurs. Cela permet de regrouper les méthodes ayant des paramètres d'entrée en commun. Suivant le fournisseur, la méthode appelée ou le centre de la scène, les paramètres d'entrées sont affichés ou non.

Si les entrées sont dynamiques (variabilité dans le temps), il est possible de visualiser ces paramètres dans une fenêtre 2D.

#### 6.4.2.4.1.JWSInputGetPanel (SINP, FMI, LATMOS, 3DView)

Cette interface permet d'appeler les méthodes getDataPointValue, getDataPointSpectra, getSurface ainsi les services de traitement 3Dview calculateDataPointValueInCube et calculateSurfaceFromCube  
Voici la liste des paramètres, et leurs caractéristiques d'apparition :

Nom	Spécificité méthode	Fournisseur	Origine ou liste des valeurs	Entrée modifiable par l'utilisateur
ResourceID			NumericalOutputID du cube	Non
Variable			ParameterKey	Non
URL x/y/z			Fichier généré par un WS 3DView	Non
IMF Clock angle		LATMOS	N'importe quel angle	Oui
Interpolation methods		FMI	NearestGridPoint, Linear	Oui
normalVector + planePoint	getSurface, calculateSurfaceFromCube		Nom de coupes prédéfinis	Oui
resolution	getSurface, calculateSurfaceFromCube			Oui

#### 6.4.2.4.2.JWSInputCalcPanel (SINP, 3DView)

Cette interface permet d'appeler les méthodes calculateDataPointValue, et calculateCube (SINP uniquement). Ces méthodes sont rajoutées comme feuille sous toutes les simulations du SINP et 3DView.

Voici la liste des paramètres, et leurs caractéristiques d'apparition :

Nom	Spécificité méthode	Origine ou liste des valeurs	Entrée modifiable par l'utilisateur	Entrée dynamique dans le temps
ResourceID		RunID	Non	Non
startTime	PointValue	Start de la scène	Non	Non
URL x/y/z	PointValue	Fichier généré par un WS 3DView	Non	Non
SW Density	SINP/Centre Earth	Appel WS AMDA	Oui	Oui
SW Pressure	3DView tsy96/Centre Earth	Appel WS AMDA	Oui	Oui
SW Velocity	Centre Earth	Appel WS AMDA	Oui	Oui
Dst	Centre Earth	Appel WS AMDA	Oui	Oui
AI	SINP, Centre Earth	Appel WS AMDA	Oui	Oui
IMF_B	Centre Earth	Appel WS AMDA	Oui	Oui
BD	Autre centre	Entrée utilisateur	Oui	Non
BT	Autre centre	Entrée utilisateur	Oui	Non
RD1	Autre centre	Entrée utilisateur	Oui	Non
RD2	Autre centre	Entrée utilisateur	Oui	Non
R1	Autre centre	Entrée utilisateur	Oui	Non
R2	Autre centre	Entrée utilisateur	Oui	Non
StepSize		0.5 radius	Oui	Non
MaxSteps/line length		1000/50 radius	Oui	Non

#### 6.4.2.4.3.JWSInputFieldLinePanel (3DView, FMI, LATMOS, SINP)

Cette interface permet d'appeler les méthodes getFieldLine.

Voici la liste des paramètres et leurs caractéristiques d'apparition :

Nom	Spécificité méthode	Fournisseur	Origine ou liste des valeurs	Entrée modifiable par l'utilisateur
ResourceID			NumericalOutputID du cube ou du SimulationRun	Non
Variable	getFieldLine		ParameterKey	Non
Direction			Forward/Backward	Oui
StepSize		FMI/LATMOS	0.5	Oui
MaxSteps		FMI	1000	Oui

#### 6.4.2.4.4.getTimeTableList et getTimeTable (AMDA, CLWeb)

Ces deux méthodes permettent de récupérer des time table d'utilisateurs Ces données sont accessibles avec un login/mot de passe qui sera soit entré au moment de l'appel, soit configuré dans les paramètres du client.

#### 6.4.2.4.5.getParameterList et getParameter (AMDA, CLWeb)

Ces deux méthodes permettent de récupérer des données d'observation publique et spécifiques à un utilisateur. Ces données sont accessibles avec un login/mot de passe qui sera soit entré au moment de l'appel, soit configuré dans les paramètres du client.

### 6.4.3.Lecture et affichage des données

En fonction du type de données sélectionné, un lecteur spécifique est lancé permettant d'extraire les données du fichier dans un tableau de valeurs. Puis un nouvel Objet est créé avec ces données permettant d'obtenir une forme graphique à partir de paramètres d'entrée comme l'échelle de couleur, une précision, ...

Ainsi, toutes les données une fois affichées auront une boîte de contrôle associée permettant de modifier certaines valeurs et changer ou régénérer la représentation graphique des données. Ces boîtes affichent toutes l'échelle de couleur avec valeur min, max et l'unité de mesure du paramètre représenté, et une option d'affichage lin/log.

Si un modèle est dynamique dans le temps, un seul objet est créé pour stocker les différentes représentations du modèle. L'objet gère les différents affichages dans le temps grâce à la méthode issue de l'interface IAnimationListener.

### 6.4.3.1. Les 3Dcubes

Ils sont fournis sous forme de fichiers NetCdf ou par un web service. Un tableau à trois dimensions, contenant la valeur du paramètre à afficher, est extrait du fichier. Puis ce cube de valeurs est transformé en couches sur les trois axes. Des QuadArray sont créés pour chaque couche et une image y est appliquée en tant que texture.

La boîte de contrôle permet de modifier le nombre de couches par axe.

L'affichage du cube permettant d'afficher des coupes.

### 6.4.3.2. Les 2Dcuts

Les coupes 2D sont au format VOTable. Le fichier XML est lu avec un parser SAX pour ne pas charger la mémoire. Un objet 3D QuadArray est créé sur lequel est appliquée une image générée à partir des données.

La boîte de contrôle permet de modifier la transparence.

### 6.4.3.3. Les TimeSeries

Les timeseries permettent d'afficher soit des vecteurs le long d'une trajectoire soit des scalaires en fonction de la feuille de l'arbre sélectionné.

Le format VOTable est lu par un parser XML SAX qui extrait des couples de valeur temps/paramètres.

La valeur temps est extraite du champ nommé « Time », le paramètre du champ nommé par la valeur extraite du « ParameterKey »

La boîte de contrôle permet de choisir la longueur des flèches, l'épaisseur du trait, la transparence et la dépendance au temps.

#### 6.4.3.4. Les Spectrogrammes

Les spectrogrammes permettent d'afficher un ruban le long d'une trajectoire.

Le format VOTable est lu par un parser XML SAX qui extrait le temps début et fin et génère une image avec en X le temps et en Y les énergies. Une trame 3D est créée sur la trajectoire avec pour point de départ et fin les temps début et fin du spectrogramme. L'image est ensuite appliquée sur cette trame.

L'affichage dynamique peut se faire en créant des « stripes » sur chaque pas et de jouer avec la transparence des « stripes » en fonction de l'animation.

La valeur temps est extraite du champ nommé « Time », le paramètre du champ nommé par la valeur extraite du « ParameterKey »

La boîte de contrôle permet de choisir la hauteur du ruban, la transparence et la dépendance au temps.

#### 6.4.3.5. Les timetables

Les timetables sont soit uploadés par l'utilisateur, soit récupérés par web service depuis AMDA et CLWeb avec les méthodes getTimeTableList et getTimeTable. Les données sont rattachées à un spacecraft de la scène. Ces données sont accessibles avec un login/mot de passe qui sera soit entré au moment de l'appel, soit configuré dans les paramètres du client.

Les timetables sont représentées en surbrillance sur la trajectoire de l'objet concerné et sont affichées indépendamment du temps de l'animation.

#### 6.4.3.6. Les lignes de champs

Les lignes de champs sont récupérées après appel à un web service. Les points d'entrée du web service sont générés grâce au web service 3DView getOrbUrl. Les points de départ des lignes sont configurés dans l'interface d'appel du web service.

Leur couleur représente la valeur du champ.

## 6.4.4. Communication spécifique avec AMDA

3DView peut s'interfacer de façon spécifique avec AMDA pour échanger des données.

### 6.4.4.1. Afficher des données d'AMDA dans 3DView

Dans le cas où 3DView est déjà lancé, AMDA peut envoyer des données sous forme de VOTable dans le Hub SAMP.

Les données transmises par AMDA sont des données science de type scalaire, vecteur, spectrogramme.

3DView dispose d'un "listener SAMP" en attente d'un éventuel appel d'AMDA. Lorsque ce module d'écoute est stimulé par la présence de données dans le hub SAMP, il demande à quel objet de la scène sont associées les données, crée les objets graphiques (scalaire sur trajectoire ou vecteur) et les affiche dans la scène.

Un vecteur n'est créé que si la première variable trouvée dans le fichier importé contient plusieurs paramètres.

Les données sont alors présentes dans la scène au même titre que celles importées depuis l'arbre de données.

## 6.4.5. Les paramètres du client

Sur la machine du client, les fichiers de configuration impex.conf et amda.conf sont dans l'espace \$HOME/.impex3dview. Le contenu de ces fichiers est modifiable via le client 3DView. Ces fichiers au format "properties" stockent des paramètres comme le login/mdp pour se connecter à AMDA et CLWeb, l'adresse des services Web, l'espace de stockage du cache, ...

## 6.4.6. Gestion du cache/espace commun

Chaque client peut gérer deux espaces : un espace commun et un cache.

### 6.4.6.1. L'espace commun

L'espace commun est un espace réseau partagé de type NFS et sert de base au service d'analyse de fichier serveur comme la génération de surface et l'interpolation de trajectoire dans un cube.

Cet espace est alimenté lors de chaque téléchargement de donnée quand il est configuré en tant qu'espace dans l'interface Interoperability/preferences.

### 6.4.6.2. Le cache

Le cache est un espace accessible directement sur la machine du client comme un accès local au disque dur. Tout fichier affiché depuis l'arbre est téléchargé dans le cache s'il n'y est pas déjà avant affichage.

Par défaut, le cache est situé dans le répertoire \$HOME/.impex3diew.