

NTIC

Projet : AT AMDA 2015

RAPPORT DES ACTIVITES LORS DE L'ASSISTANCE TECHNIQUE AMDA 2015

Rédigé par : Benjamin Renard	Diffusé à : CDPP / IRAP
Approuvé par :	

LISTE DES MODIFICATIONS DU DOCUMENT

Vers.	Date	Paragraphe	Description de la modification
01.0			Création du document

SOMMAIRE

1	INTRODUCTION	6
2	MISE EN PLACE DE LA PLATEFORME DE DEVELOPPEMENT POUR LE MODULE AMDA_KERNEL SUR LA MACHINE « APUS »	7
3	MERGE DES SOURCES	8
4	MISE EN PLACE DES REPOSITORIES GIT POUR L'APPLICATION AMDA.....	9
5	ETUDE DE MIGRATION DE L'IHM AMDA DE EXTJS 4.1.3 VERS EXTJS 4.2.2	10
5.1	Points majeurs.....	10
5.1.1	Erreurs de type « Ressources manquantes »	10
5.1.2	Erreurs de positionnement des fenêtres flottantes « outils ».....	11
5.1.3	Erreurs du module « Explorer »	12
5.1.4	Suppression d'un élément dans une grille	17
5.2	Points mineurs.....	17
5.2.1	Curseur de la souris lors du survol des nœuds de l'explorer.....	17
5.2.2	Taille des items du menu « start »	18
5.2.3	Taille du contenu des différentes fenêtres d'aide	18
6	ETUDE LA MIGRATION DE L'IHM AMDA DE EXTJS 4.2.2 VERS EXTJS 4.2.4	19
6.1.1	Erreur de construction de l'arbre de l'Explorer	19
6.1.2	Erreur de construction de l'arbre du module « Plot »	19
6.1.3	Erreur lors de la définition d'un « scatter plot » dans le module « Plot ».....	20
7	CHARGEMENT DYNAMIQUE DES MODULES DE AMDA_IHM.....	21
7.1	Etat des lieux.....	21
7.2	Lecture dynamique des modules	22
7.2.1	Mise en place générale.....	22
7.2.2	Chargement réel des modules.....	24
8	GESTION DES GRANDES TIME TABLES.....	26

9	VALIDATON DES NOMS DES OBJETS COTE SERVEUR.....	27
10	UTILISATION DE L'ADRESSE IP UTILISATEUR POUR LES ECHANGES ENTRE AMDA_KERNEL ET DDSEVER.....	28
11	CREATION D'UN REPOSITORY POUR DDSEVER ET COMPILATION AVEC CMAKE.....	29
12	ACCES CONCURRENTS A DDSEVER.....	30
13	ETUDE ET CORRECTION DU PROBLEME DE PERFORMANCE LORS DE L'APPEL A DD_GETMULTIDATA.....	31
14	COMPILATION DES PARAMETRES DERIVES AU MOMENT DE LEUR SAUVEGARDE.....	32
15	MODIFICATION DE LA PALETTE DE COULEURS UTILISEE POUR LES PLOTS.....	33
16	AMELIORATION DE L'AFFICHAGE INTERACTIF DES PLOTS.....	34
17	ALIGNEMENT DES « PLOTS AREA » AVEC LE « LAYOUT » MANUEL.....	35
18	INFORMATION DE CALIBRATION POUR PARAMETRES ISSUS DE FICHIERS LOCAUX.....	36
18.1	Définition des informations de calibration au niveau du module AMDA_Kernel.....	36
18.1.1	Définition d'une information de calibration liée au ParamGet LocalFileInterface.....	36
18.1.2	Définition d'une information de calibration de manière manuelle (non liée à un ParamGet).....	36
18.2	Prise en compte des informations de calibration au niveau du module AMDA_Kernel.....	36
18.2.1	Prise en compte d'une information de calibration liée au ParamGet LocalFileInterface.....	36
18.2.2	Prise en compte d'une information de calibration « manuelle ».....	37
18.3	Définition 'une table au niveau du module AMDA_IHM.....	37
19	PRISE EN COMPTE DES MODIFICATIONS DU SERVICE DE LA BASE APIS DANS AMDA.....	38
20	PRISE EN COMPTE DES FORMATS DE TEMPS EPOCH16 ET TT2000 DANS L'APPLICATION « CONVERT2NC ».....	39
21	MODIFICATIONS AUTOUR DE L'APPLICATION TREPS.....	40
21.1	Rappel sur la mauvaise gestion des NaN par le Webservice 3DView.....	40
21.2	Ouverture de l'aide à l'entrée de l'application.....	40
21.3	Amélioration du mécanisme de conversion du champ « temps » d'un fichier source.....	40
21.4	Précision des nombres flottants dans le cas d'un fichier source ASCII.....	40



**Rapport des activités lors de l'Assistance Technique AMDA
2015**

22	DOCUMENTS APPLICABLES ET DE REFERENCE (A/R).....	42
23	GLOSSAIRE ET ABREVIATIONS.....	43
23.1	Glossaire.....	43
23.2	Abréviations.....	43

1 INTRODUCTION

La prestation « Assistance Technique AMDA 2015 » s'est déroulée du lundi 20 avril 2015 au vendredi 3 juillet 2015 (44 jours de charge, pour un ETP AKKA) dans les locaux de l'IRAP.

Cette prestation concernait un ensemble de tâches diverses, presque toutes liées au projet AMDA du CDPP / IRAP, référencées dans la proposition commerciale [R0]. Il est à noter que des tâches supplémentaires ont été traitées au cours de la prestation, en accord avec l'équipe technique du CDPP / IRAP.

Ce document constitue un rapport de l'ensemble des activités réalisées lors de cette prestation.

2 MISE EN PLACE DE LA PLATEFORME DE DEVELOPPEMENT POUR LE MODULE AMDA KERNEL SUR LA MACHINE « APUS »

La machine « apus » est la machine de développement mise à disposition par l'IRAP au cours de la prestation.

Cette machine fonctionne avec un système d'exploitation de type « Linux - Fedora 19 - 64 bits », alors que le module « AMDA_Kernel » a été développé pour un système d'exploitation de type « Linux – CentOS 6.3 – 64 bits », cf. [R1].

Dans l'ensemble, la compilation des COTS et du module n'a pas posé de gros problèmes.

Toutefois, au moment de l'installation de GCC et de boost (cf. [R2], §5.3), c'est-à-dire au moment de l'exécution du script « bld.sh », une fois que les sources sont décompressées par le script et que la compilation a débuté, il faut être suffisamment réactif afin de procéder à la modification suivante :

⇒ Dans le fichier « /opt/tools/gcc/4.7.2/src/libiconv-1.14/src/lib/stdio.in.h », la ligne 698 est supprimée et devient :

```
#if defined(__GLIBC__) && !defined(__UCLIBC__) && !__GLIBC_PREREQ(2, 16)
  _GL_WARN_ON_USE (gets, "gets is a security hole – use fgets instead");
#endif
```

Source: <https://forum.directadmin.com/showthread.php?t=48128>

Attention, les images générées par le ParamOutput « Plot » du module AMDA_Kernel sont légèrement différentes sur la machine « apus » que celles générées sur une CentOS 6.3. Ces différences sont très probablement dues au fait que les versions des polices de caractères et de la librairie « libpng » diffèrent entre ces deux machines.

Ainsi, lors de l'exécution des tests « Fitnessse », toutes les requêtes de type « Plot » apparaissent en échec.

La validation réelle ne pourra donc se faire que sur la machine CentOS.

3 MERGE DES SOURCES

Plusieurs branches des sources de l'application AMDA, avec des développements spécifiques pour chacune d'entre elles, cohabitaient. On peut notamment référencer :

- La version de développement de l'IRAP, gérée par SVN (svn://cdpp1.cesr.fr/depot/AMDA-NG4/trunk). Cette branche contient les sources des derniers développements de l'IHM réalisés par l'IRAP, et l'ensemble des scripts PHP d'intégration avec l'ancien noyau de l'application,
- La version « industrialisée » telle que livrée à l'issue de la troisième prestation d'industrialisation du nouveau noyau AMDA. Il s'agit de la version 3.3.0 du module AMDA_Kernel (nouveau noyau d'AMDA) et de la version 1.2.0 du module AMDA_Integration. Ces modules ont été développés afin de fonctionner avec la version 1.3.1 de l'IHM.

Cette tâche consistait à merger ces sources afin d'obtenir l'application:

- Contenant les derniers développements sur l'IHM réalisés par l'IRAP,
- Fonctionnant avec le module AMDA_Kernel version 3.3.0,
- Fonctionnant avec le module AMDA_Integration version 1.2.0.

Les sources mergées de l'application sont dorénavant gérées sous GIT (cf. §4).

4 MISE EN PLACE DES REPOSITORIES GIT POUR L'APPLICATION AMDA

L'équipe technique du CDPP / IRAP souhaite gérer les versions de ses projets sous GIT.

Au sein du laboratoire, un « gitlab » a été mis en place et a donc été utilisé pour AMDA (<https://gitlab.irap.omp.eu>).

Il n'a pas été demandé de réaliser un import complet des repositories SVN existants vers GIT, avec les logs de modification des fichiers sources. La méthodologie suivante a été adoptée :

- Un export des sources des repositories SVN a été effectué,
- Sur ces sources, l'opération de merge (cf. §3) a été réalisée,
- 3 nouveaux repositories GIT ont été créés :
 - https://gitlab.irap.omp.eu/CDPP/AMDA_IHM.git, pour le module AMDA_IHM
 - https://gitlab.irap.omp.eu/CDPP/AMDA_Integration.git, pour le module AMDA_Integration,
 - https://gitlab.irap.omp.eu/CDPP/AMDA_Kernel.git, pour le module AMDA_Kernel.
- Les sources mergées ont été committées dans ces repositories GIT.

Les principales commandes GIT utilisées sont :

- « git clone REPOSITORY » : pour créer un repository local de REPOSITORY,
- « git status » : pour obtenir la liste des modifications sur les fichiers sources,
- « git add FICHIER » : pour ajouter « FICHIER » dans la liste des fichiers candidats au commit. Remarque :
« git add -A » : permet d'ajouter tous les fichiers modifiés avec une seule commande,
- « git commit » : pour réaliser un commit local de la liste des fichiers candidats au commit,
- « git push » : pour pousser le travail contenu dans le repository local vers le dépôt distant Gitlab,
- « git diff FICHIER » : pour voir les modifications de « FICHIER » depuis le dernier commit.

5 ETUDE DE MIGRATION DE L'IHM AMDA DE EXTJS 4.1.3 VERS EXTJS 4.2.2

Les différents points relevés lors de cette étude ont été catégorisés de la manière suivante :

- Les points majeurs : il s'agit de points bloquants, ou des bugs de fonctionnement, dont la non-résolution empêche le bon fonctionnement de l'application, ou risque d'altérer l'espace de travail de l'utilisateur. Ces différents points sont détaillés de la manière suivante :
 - Constat : Description succincte du problème rencontré et de ses conséquences sur l'application,
 - Explication : Description détaillée de l'origine du problème soulevé, désignation du code incriminé, etc...
 - Solution : Proposition de correction du problème.

Les points mineurs : il s'agit de points non bloquants, et probablement non gérants pour l'utilisateur. Généralement, il s'agit de différences d'affichages de l'application entre les deux versions du framework, n'empêchant pas son utilisation.

5.1 POINTS MAJEURS

5.1.1 Erreurs de type « Ressources manquantes »

5.1.1.1 Fichier « CheckHeader.css »

5.1.1.1.1 Constat

La ressource CSS « js/lib/ext/examples/ux/css/CheckHeader.css » est manquante et provoque un échec du chargement de l'application.

5.1.1.1.2 Explication

Cette ressource CSS n'est utilisée que par le composant « Ext.ux.CheckColumn », lui-même utilisé par le script « js/app/views/InteroplmpexUI.js » de l'application AMDA.

D'après les « release notes » de la version 4.2.2 du framework ExtJS, le composant « Ext.ux.Checksum » fait désormais partie intégrante du framework (cf. ticket EXTJSIV-8539) avec le nom suivant « Ext.grid.column.CheckColumn ».

Un mapping existe faisant le lien entre l'ancien nom du composant « Ext.ux.CheckColumn » et le nouveau nom « Ext.grid.column.CheckColumn » (cf. « ext-debug.js », ligne 22338).

Les ressources CSS associées à ce composant sont donc désormais automatiquement chargées lors du chargement du framework.

5.1.1.1.3 Solution

Le chargement de cette ressource étant assuré par le framework, elle a été retirée du fichier source « desktop.php » de l'application AMDA.

5.1.1.2 Fichier « PagingScroller.js »

5.1.1.2.1 Constat

La ressource Javascript « js/lib/ext/src/grid/PagingScroller.js » est manquante et provoque un échec du chargement de l'application.

5.1.1.2.2 Explication

L'appel du chargement de cette ressource est effectué par le script « js/app/views/TimeTableUI.js ».

Cette ressource est manquante dans la version 4.2.2 du framework.

Si le chargement du script est demandé, il n'est en revanche pas utilisé par la suite dans l'application AMDA.

De plus, les commentaires associés à cette ressource dans la version 4.1.3 du framework précise que : « this class should never be instanciated by user code ».

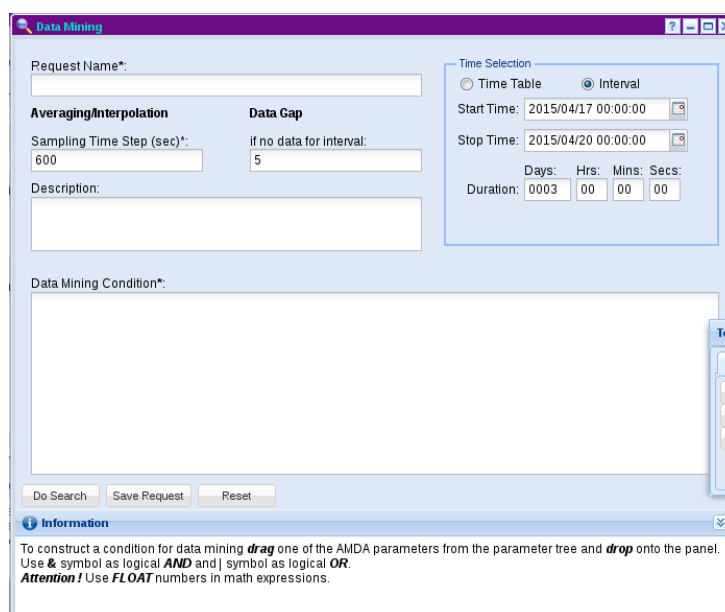
5.1.1.2.3 Solution

Compte tenu du fait de la non utilisation de ce composant par l'application AMDA, le chargement de la ressource a été retirée du script « js/app/views/TimeTableUI.js ».

5.1.2 Erreurs de positionnement des fenêtres flottantes « outils »

5.1.2.1 Constat

Les fenêtres flottantes « outils » associées à certains modules de l'application AMDA n'apparaissent pas à la bonne position.



De plus, de mauvaises contraintes de positionnement sont appliquées, empêchant le repositionnement des fenêtres volantes par l'utilisateur.

Ce problème se produit :

- Pour la fenêtre « Tools For Parameter Construction » du module « Create/modify parameters »,
- Pour la fenêtre « Tools For Condition Construction » du module « Data Mining »,
- Pour la fenêtre « Statistical info » du module « Manage Time Tables »,
- Pour la fenêtre « Select Arguments for ... » du module « Plot Manager »,
- Pour la fenêtre « Select arguments for Panel ... » du module « Plot Manager »,
- Pour la fenêtre « Interactive interval » du module « Plot result ».

5.1.2.2 Explication

Les contraintes de repositionnement s'appliquent sur les positions absolues des fenêtres flottantes, et non pas sur les positions relatives à la fenêtre parente.

A priori, il s'agit d'un bug d'ExtJS 4.2.2.

5.1.2.3 Solution

Les fonctions « getConstrainVector » de chacune de ces fenêtres flottantes sont surchargées de manière à reproduire le comportement de la version 4.1.3 d'ExtJS :

```
getConstrainVector: function(constrainTo){  
    var me = this;  
    if (me.constrain || me.constrainHeader) {  
        constrainTo = constrainTo || (me.floatParent && me.floatParent.getTargetEl()) || me.container ||  
me.el.getScopeParent();  
        return (me.constrainHeader ? me.header.el : me.el).getConstrainVector(constrainTo);  
    }  
}
```

5.1.3 Erreurs du module « Explorer »

5.1.3.1 Actions des menus contextuels non exécutés

5.1.3.1.1 Constat

Un clic droit sur un nœud du module « Explorer » provoque l'apparition d'un menu contextuel contenant un ensemble d'actions relatives au nœud sélectionné.

Lors d'un clic sur une de ces actions, le menu contextuel disparaît mais l'action n'est plus exécutée.

5.1.3.1.2 Explication

A l'issue de la création du menu contextuel associé à la sélection du module « Explorer », deux événements sont instanciés au niveau du menu :

- Un événement « click », dispatchant les actions vers les enregistrements sélectionnés,
- Un événement « beforehide », supprimant l'événement « click » du menu lors de la disparition du menu contextuel.

Dans ExtJs, l'appel à ces événement est asynchrone, il n'est donc pas possible de garantir l'ordre d'exécution de ces deux événements.

Il se trouve qu'avec la migration, l'événement « beforehide » est appelé avant l'événement « click ». Ainsi, l'événement « click » est supprimé avant même qu'il ne soit exécuté.

5.1.3.1.3 Solution

La suppression de l'événement « click » doit se faire dans la fonction exécuté par cet événement afin de garantir le bon ordre d'exécution.

```
var onRecordClick = function (menu, item, e, eOpts) {  
    //Dispatch click event to the record  
    this.onMenuItemClick(menu,item,e);  
    //Remove old click listener  
    menu.removeListener('click',onRecordClick,this);  
};  
menu.addListener('click',onRecordClick,rec);
```

5.1.3.2 *Exception lors de l'édition d'un nom d'un répertoire*

5.1.3.2.1 Constat

Une exception « context.view is undefined » survient dans la fonction « getEditingContext » du composant « Ext.grid.plugin.Editing ».

5.1.3.2.2 Explication

Le contexte d'exécution d'une opération d'édition est fourni par l'appel à la fonction « getEditingContext » du composant « Ext.grid.plugin.Editing ».

Cette fonction est surchargée dans la définition de l'objet « MyTreeEditor » dans le script « js/app/views/ExplorerUI.js ».

Le contenu du contexte a été enrichi dans la version 4.2.2 du framework, et nécessite notamment la définition du « view ».

5.1.3.2.3 Solution

Le retour de la fonction « getEditingContext » définit dans le script « js/app/views/ExplorerUI.js » devient :

```
return {  
  grid: grid,  
  record: record,  
  field: columnHeader.dataIndex,  
  value: value,  
  column: columnHeader,  
  colIdx: colIdx,  
  view: columnHeader.getOwnerHeaderCt().view  
};
```

5.1.3.3 Exception lors du rafraichissement de l'arbre suite à une édition du nom d'un répertoire

5.1.3.3.1 Constat

Une exception « column is null » dans la fonction « shouldUpdateCell » du composant « Ext.view.Table » survient lors du rafraichissement de l'arbre après une opération d'édition du nom d'un répertoire.

5.1.3.3.2 Explication

La fonction « shouldUpdateCell » est appelée par la fonction « onUpdate » du composant « Ext.view.Table ».

Cette fonction « onUpdate » est surchargée dans le script « js/lib/ext-override.js ».

Par ailleurs, un commentaire précise : « TODO check if needed for version 4.1.3, 4.2.2 ».

5.1.3.3.3 Solution

Le bug corrigé par la surcharge de cette méthode semblant être résolu dans la version 4.2.2 du framework, la suppression de cette surcharge est proposée.

5.1.3.4 Exception lors de l'annulation d'une opération d'édition d'un nom d'un répertoire

5.1.3.4.1 Constat

Lorsqu'un utilisateur annule une opération d'édition d'un nom d'un répertoire, une exception « this.get(...) is undefined » survient au niveau du constructeur du composant « InteractiveNode ».

5.1.3.4.2 Explication

L'édition des nœuds de l'explorer est gérée par le plugin « MyTreeEditor » défini dans le script « js/app/views/ExplorerUI.js ».

Ce plugin implémente notamment la fonction « cancelEdit » appelée à l'annulation d'une édition d'un nom d'un répertoire.

Il semblerait que le record correspondant soit supprimé avant que l'annulation de l'édition ne soit complètement effective, provoquant ensuite cette exception.

5.1.3.4.3 Solution

Le patch suivant est proposé.

La fonction « cancelEdit » du composant « MyTreeEditor » devient :

```
cancelEdit: function() {  
    var me = this,  
    activeEd = me.getActiveEditor(),  
    viewEl = me.grid.getView().getEl(me.getActiveColumn());  
  
    me.setActiveEditor(null);  
    me.setActiveColumn(null);  
    me.setActiveRecord(null);  
    if (activeEd) {  
        activeEd.cancelEdit();  
        viewEl.focus();  
        this.fireEvent('canceledit', activeEd, me.context);  
    }  
}
```

Dans ce patch, ce n'est plus la fonction « canceledit » qui supprime le nœud. En revanche, elle déclenche un événement « canceledit » au niveau de l'éditeur.

L'instanciation du plugin devient :

```
plugins: [new MyTreeEditor({  
    pluginId: 'treeEditor',  
    listeners: {  
        'canceledit': function(editor, context) {  
            if (editor && editor.startValue===amdaModel.AmdaNode.NEW_DIR_NAME) {  
                context.record.remove(true);  
            }  
        }  
    }  
})]
```

5.1.3.5 Mauvais menu contextuel

5.1.3.5.1 Constat

En arrivant sur le bureau d'AMDA, et réalisant directement un clic droit sur le nœud « My Times Tables », le menu contextuel qui apparaît correspond à celui d'une sélection multiple de Time Tables.

5.1.3.5.2 Explication

Dans la fonction « itemcontextuelmenu » du composant « ExplorerUI », un test est effectué sur le modèle de sélection afin de savoir s'il s'agit d'une sélection multiple ou non.

En fonction de ce test, le menu contextuel correspondant est censé être construit.

Hors le test est fait de telle manière que si la sélection est vide, le menu contextuel de sélection construit est celui du cas « multi sélection », ce qui est un premier bug.

Ensuite, le modèle de sélection ne semble pas être mis à jour lors du premier clic droit dans l'explorer (uniquement si aucune sélection n'a auparavant été faite dans l'arbre), ce qui est un second bug.

5.1.3.5.3 Correction

Le patch suivant est proposé :

Tout d'abord, il faut forcer l'ajout du record courant dans le model de sélection dans la fonction « itemcontextuelmenu », afin de s'assurer qu'il soit bien à jour par rapport à la sélection dans l'arbre :

```
view.ownerCt.getSelectionModel().select(rec,true);
```

Ensuite, il faut corriger le test sur le type de la sélection courante :

```
if (view.ownerCt.getSelectionModel().selected.length === 1) {  
    // get items menu corresponding to right clicked record  
    menuItems = rec.getContextMenuItems(this);  
} else if (view.ownerCt.getSelectionModel().selected.length > 1) {  
    // get items menu corresponding to right clicked record  
    menuItems = rec.getContextMenuMultiItems(this);  
}
```

5.1.3.6 Erreurs multiples lors d'une multi-sélection et de l'ouverture du menu contextuel avec un clic droit

5.1.3.6.1 Constat

Plusieurs erreurs ont été détectées lors de l'ouverture du menu contextuel avec un clic droit, lors d'une sélection multiple.

5.1.3.6.2 Explication

Tout d'abord, le composant « ExplorerUI » utilise l'option « multiSelect » qui est obsolète.

De plus, l'utilisation de cette option rentrait potentiellement en conflit avec la définition du type de model de sélection :

```
selModel: { //use the grid ROW model  
    selType: 'rowmodel'  
}
```

5.1.3.6.3 Solution

L'option « multiSelect » ne doit plus être utilisée.

Ensuite, il faut définir le type de modèle de sélection dédié aux arbres, avec les options permettant la multi-sélection et gérant l'utilisation du clic droit pour l'ouverture du menu contextuel, soit :

```
selModel: Ext.create('Ext.selection.TreeModel', {  
    ignoreRightMouseSelection: true,  
    mode: 'MULTI'  
}))
```


5.1.4 Suppression d'un élément dans une grille

5.1.4.1 Constat

La suppression d'un élément d'une grille s'effectue sur la ligne qui a le focus, même si elle est demandée sur une autre ligne.

Ce problème se produit sur :

- La grille du module « Operations on Time tables »,
- La grille du module « Download Data »,
- La grille du « TimeSelector ».

5.1.4.2 Explication

Le modèle de sélection de la grille n'est pas forcément à jour au moment du déclenchement de l'événement « cellclick ».

5.1.4.3 Solution

La fonction associée à l'événement est modifiée de la manière suivante :

```
this.TTGrid.on('cellclick', function(view, cell, cellIndex, record, row, recordIndex, e){  
    if (cellIndex == 2)  
        this.TTGrid.getStore().remove(record);  
},this);
```

5.2 POINTS MINEURS

5.2.1 Curseur de la souris lors du survol des nœuds de l'explorer

Le curseur de la souris lors du survol d'un nœud est de type « default » avec la version 4.2.2 du framework, alors qu'il était de type « pointer » avec la version 4.1.3.

La représentation du curseur peut être modifiée afin de revenir au comportement précédent. Dans la fonction « itemmouseenter » du composant « ExplorerUI » :

```
itemmouseenter: function(view, record, item){  
    if(record.get('isParameter')){  
        var el = Ext.get(item),  
            td = el.down('td > div');  
        td.setStyle('cursor', 'crosshair');  
    }  
    else  
    {  
        var el = Ext.get(item),  
            td = el.down('td > div');  
        td.setStyle('cursor', 'pointer');  
    }  
}
```

5.2.2 Taille des items du menu « start »

La hauteur des items contenus par le menu « Start » est légèrement plus petite avec la version 4.2.2 du framework qu'avec la version 4.1.3.

Il s'agit probablement d'une modification des feuilles de styles du framework.

5.2.3 Taille du contenu des différentes fenêtres d'aide

Des différences de taille des différents éléments HTML tracés dans les pages d'aide de l'application ont été détectées.

Il s'agit probablement d'une modification des feuilles de styles du framework.

6 ETUDE LA MIGRATION DE L'IHM AMDA DE EXTJS 4.2.2 VERS EXTJS 4.2.4

En cours de prestation, et notamment lors de la mise en place du mécanisme de gestion des grosses TimeTables (cf. §8), des bugs ExtJS ont été détectés. Ces bugs sont corrigés dans la version 4.2.4 du framework (release non publique au moment de la prestation), justifiant ainsi cette migration.

6.1.1 Erreur de construction de l'arbre de l'Explorer

6.1.1.1 Constat

Les nœuds constituant l'arbre de l'Explorer ne s'affichent plus.

6.1.1.2 Explication

Un nœud d'un arbre ExtJS doit être une instance d'un Model décorée avec un NodeInterface.

« amdaModel.AmdaNode » hérite de « Ext.data.Model » mais n'est pas décoré avec un NodeInterface. Dans ces conditions, le nœud ne peut être correctement affiché.

6.1.1.3 Solution

La classe « amdaModel.AmdaNode » doit hériter de « Ext.data.TreeModel ».

6.1.2 Erreur de construction de l'arbre du module « Plot »

6.1.2.1 Constat

Les nœuds constituant l'arbre du module « Plot » ne s'affichent plus.

6.1.2.2 Explication

Un nœud d'un arbre ExtJS doit être une instance d'un Model décorée avec un NodeInterface.

Un nœud du module « Plot » est une instance d'un « amdaModel.PlotObject », héritant de « amdaModel.AmdaObject » héritant lui-même de « Ext.data.Model », il n'est donc pas décoré avec un NodeInterface. Dans ces conditions, le nœud ne peut être correctement affiché.

6.1.2.3 Solution

La décoration du nœud est forcée de la manière suivante :

```
Ext.define('amdaModel.PlotObject', {  
    //...  
}, function () {  
    Ext.data.NodeInterface.decorate(this);  
});
```

6.1.3 Erreur lors de la définition d'un « scatter plot » dans le module « Plot »

6.1.3.1 Constat

Il n'est plus possible de définir un « scatter plot » dans le module « Plot ».

En effet, lorsque l'on dépose la composante X sur le paramètre Y (après avoir sélectionné le type « SCATTER »), le nœud déposé se retrouve enfant du nœud « Panel ».

6.1.3.2 Explication et Solution

L'interface du module « Plot » est complexe et a déjà subi plusieurs migrations (notamment de ExtJS 3 vers ExtJS 4). Il semblerait que l'ensemble du mécanisme de définition d'un « scatter plot » ait besoin d'être corrigé.

Ainsi, et compte tenu qu'un rework de ce module est prévu, il a été décidé de ne pas corriger cette erreur pour l'instant.

7 CHARGEMENT DYNAMIQUE DES MODULES DE AMDA_IHM

7.1 ETAT DES LIEUX

L'IHM de l'application AMDA est architecturée autour d'un modèle de type MVC, apportant une certaine clarté à l'application.

Le chargement des modules est géré au niveau de la couche « contrôleur » de l'application, notamment par la classe « AmdaApp » héritant du composant ExtJS « Ext.ux.desktop.App ».

Le diagramme de classe suivant montre les classes mises à contribution pour le chargement de l'application.

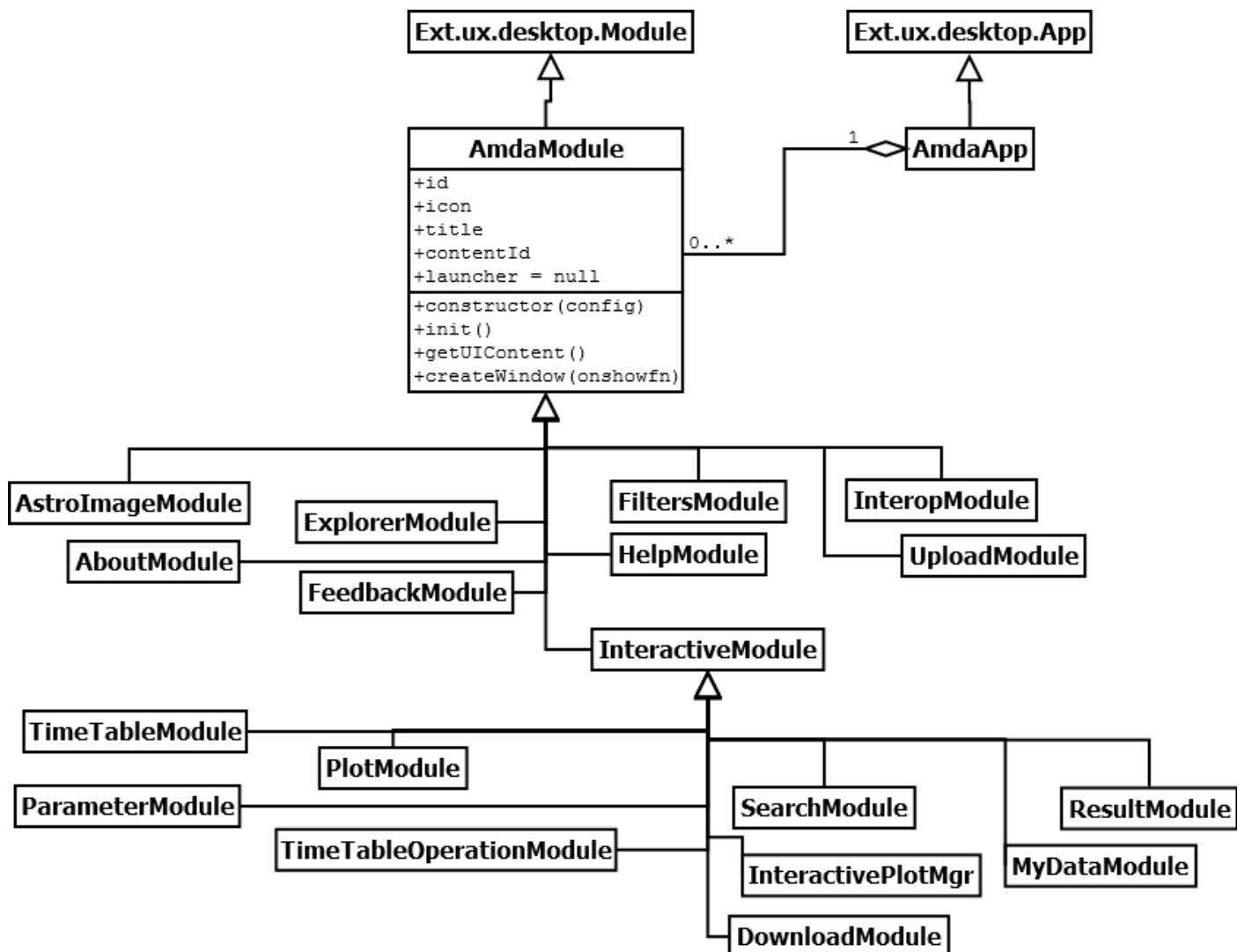


Figure 1 - Diagramme de classe de la partie « chargement des modules »

Un module AMDA est porté par une classe héritant de la classe « AmdaModule » héritant elle-même du composant ExtJS « Ext.ux.desktop.Module ».

Les différents modules de l'application sont agrégés par le singleton « AmdaApp ».

Le diagramme de séquence suivant montre le mécanisme d'initialisation et de chargement des modules par le singleton « AmdaApp » :

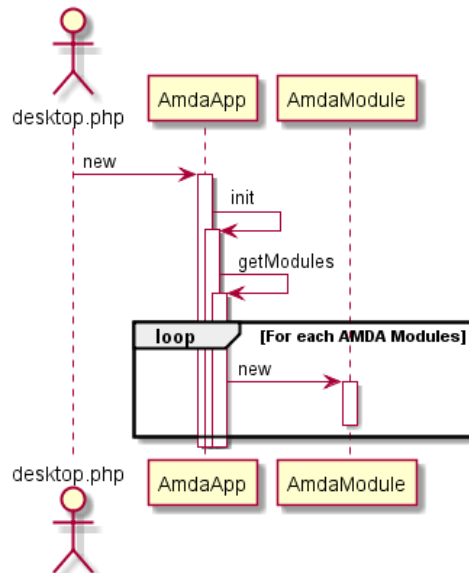


Figure 2 - Diagramme de séquence d'initialisation et de chargement des modules

Comme le montre le diagramme précédent, chaque module est chargé et instancié par le système dès la création du singleton « AmdaApp », lui-même créé par le script PHP « desktop.php ».

Ainsi tous les modules, ainsi que leurs dépendances, sont chargés et instanciés dès l'entrée de l'utilisateur dans l'application.

7.2 LECTURE DYNAMIQUE DES MODULES

7.2.1 Mise en place générale

Comme montré dans la section §7.1, tous les modules sont chargés et instanciés dès l'entrée de l'utilisateur dans l'application.

Afin d'optimiser le chargement global de l'application, il a été demandé de réaliser un chargement dynamique « à la demande » des modules.

Le nouveau diagramme de classe correspondant au chargement de l'application devient :

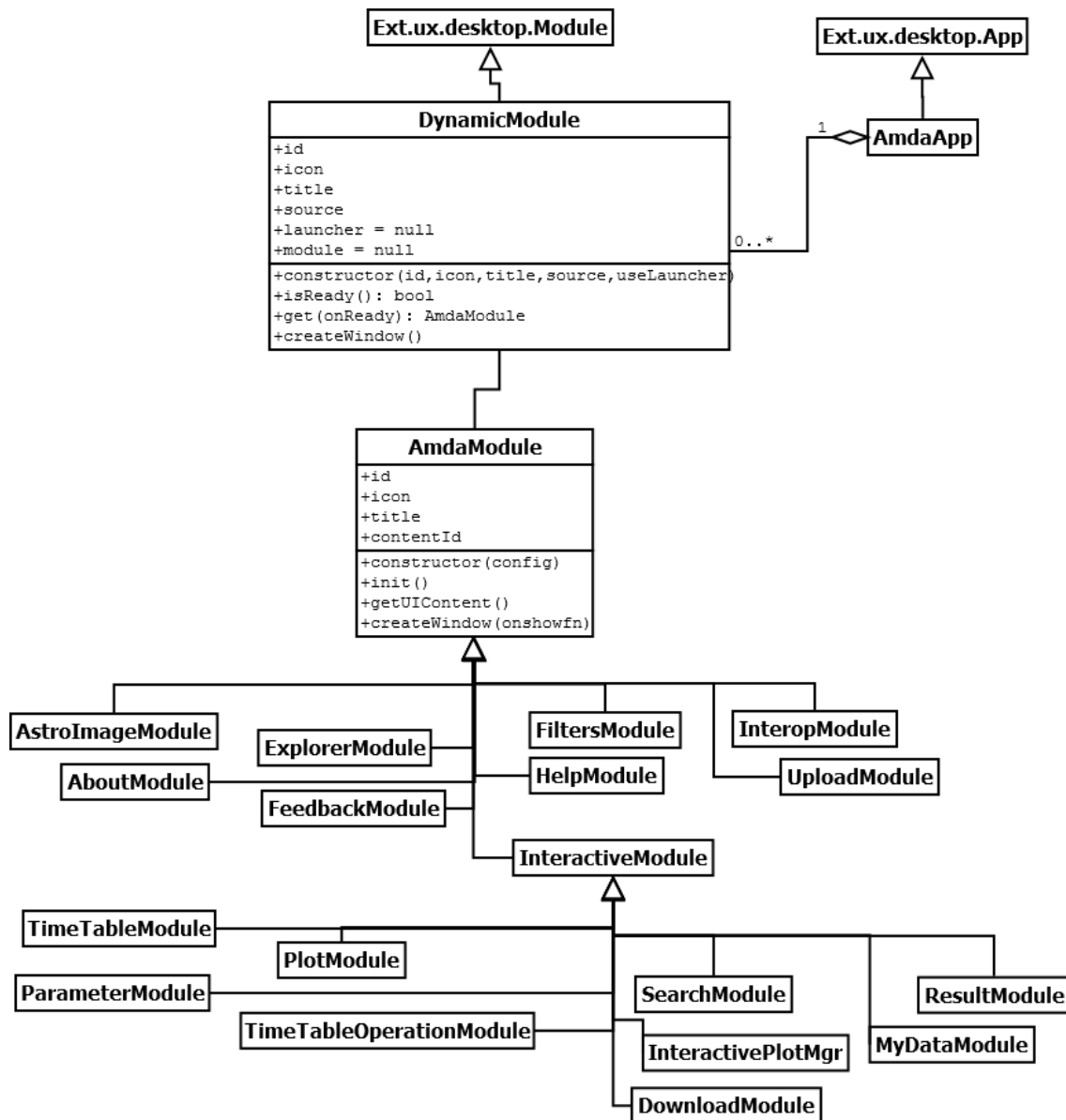


Figure 3 - Diagramme de classe de la partie « chargement dynamique des modules »

La classe « DynamicModule » a été introduite. Elle hérite de la classe « Ext.ux.desktop.Module », et ne contient que les éléments indispensables à la création du « desktop » :

- L'attribut « id » représente l'identifiant unique du module,
- L'attribut « icon » désigne l'icône associée au module,
- L'attribut « title » désigne le titre du module,
- L'attribut « source » désigne le script source correspondant au module (par exemple « amdaDesktop.AboutModule » pour le module « AboutModule »),
- L'attribut « launcher » qui est une structure utilisée lorsque le module est affiché dans le menu « Start » de l'application,

- L'attribut « module » représentant l'instance chargée du module (donc à « null » tant que le module n'a pas été réellement chargé).

Dorénavant, à la création du singleton « AmdaApp », seule une instance d'un « DynamicModule » est créée pour chaque module d'AMDA, comme montré par le diagramme de séquence suivant :

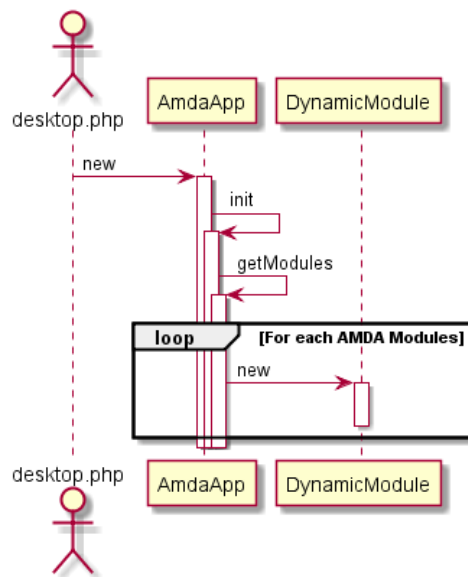


Figure 4 - Diagramme de séquence d'initialisation des modules dynamiques

Ainsi, les modules sont bien initialisés au niveau du « Desktop », sans pour autant charger toutes les dépendances qui leurs sont associées.

7.2.2 Chargement réel des modules

Les modules réels sont chargés à la demande.

La séquence suivante décrit le mécanisme mis en œuvre lorsqu'un utilisateur demande l'affichage d'une fenêtre d'information :

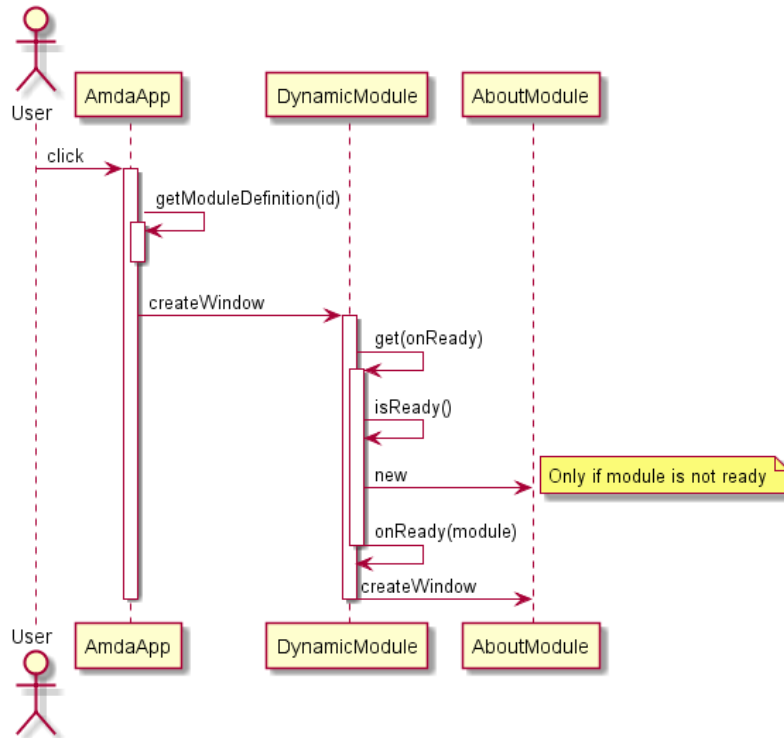


Figure 5 - Diagramme de séquence pour ouverture d'une fenêtre d'information

8 GESTION DES GRANDES TIME TABLES

Le chargement d'une TimeTable contenant un grand nombre d'intervalles de temps posait problème au niveau de son affichage dans AMDA_IHM.

Si ExtJS prévoit une option de « bufferisation » pour l'affichage d'une grille avec un grand nombre d'enregistrements, elle n'est en revanche pas compatible avec le mode édition (ajout/suppression/modification d'intervalles).

Pour contourner ce problème, tous les traitements sur les Time Tables lors d'une édition sont effectués côté serveur. Pour se faire, un fichier « cache » est créé dans l'espace de l'utilisateur afin de contenir l'état de sa Time Table pendant son édition.

Ce fichier « cache » est dans un format binaire, permettant un accès rapide aux données lors des échanges avec l'IHM. Il est géré par la nouvelle classe PHP « TimeTableCacheMgr ».

Ainsi, le composant grille ExtJS peut être utilisé en « lecture seule », et donc avec l'option de bufferisation, son contenu étant rechargé à chaque modification du cache de la Time Table.

Dorénavant, les opérations de récupération des statistiques, et de merge/extend/shift/tri/filtrage des intervalles, s'effectuent tous « côté serveur » et sont gérés par la classe « TimeTableCacheMgr ».

9 VALIDATON DES NOMS DES OBJETS COTE SERVEUR

La fonction « validNameObject » a été introduite au niveau de la classe « AmdaAction ».

Cette fonction est appelée, via Ext.Direct, lors de la validation des noms des objets : paramètres uploadés, paramètres dérivés, Time Table, condition d'un Data Mining, requête enregistrée.

10 UTILISATION DE L'ADRESSE IP UTILISATEUR POUR LES ECHANGES ENTRE AMDA KERNEL ET DDSERVER

La propriété « app.user.host » a été introduite dans le fichier de configuration « app.properties » du module AMDA_Kernel, afin de définir l'adresse IP de l'utilisateur à l'origine de la requête.

Lorsque cette propriété est définie, elle est fournie à la classe « DD_Client » par le biais de la fonction « setUserHost ». Dans ces conditions, c'est cette adresse IP qui est envoyée vers DDServer lors des requêtes, et non pas l'adresse IP du serveur hébergeant l'application AMDA.

A chaque requête traitée par le module AMDA_Integration, un fichier de configuration « app.properties » est généré, contenant notamment la propriété « app.user.host » initialisée avec l'adresse IP de l'utilisateur à l'origine de la requête.

11 CREATION D'UN REPOSITORY POUR DDSERVER ET COMPILATION AVEC CMAKE

Un repository GIT spécifique a été créé pour l'application DDServer : <https://gitlab.irap.omp.eu/CDPP/DDServer.git>.

La compilation s'effectue à partir de CMake. La procédure de compilation et d'installation est contenue dans le fichier « README » du projet.

Il est à noter que les liens vers les librairies netCDF et DD sont dorénavant des liens dynamiques.

12 ACCES CONCURRENTS A DDSERVER

Pour le moment, afin de gérer correctement (mais pas de manière optimale) les accès concurrents à DDServer, un système de lock des VirtualInstrument a été mis en place.

Ainsi, un seul client peut accéder à un VirtualInstrument donné à la fois.

Les méthodes de « lock » et de « unlock » sont définies dans le fichier « DD_CacheLock.c ». Ces méthodes utilisent des « mutex », définis dans un espace mémoire partagé entre les différents process, pour bloquer ou débloquer l'accès à un VirtualInstrument.

A l'avenir, il serait intéressant d'optimiser ce traitement en limitant autant que possible le blocage du VirtualInstrument (pour le moment il est bloqué sur toute la durée du traitement d'une requête de récupération de données).

13 ETUDE ET CORRECTION DU PROBLEME DE PERFORMANCE LORS DE L'APPEL A DD_GETMULTIDATA

Un problème de performance avait été détecté par l'IRAP qui se manifestait par un temps d'exécution particulièrement élevé des appels à la fonction « DD_GetMultiData » par le module AMDA_Kernel.

Après prospection, nous nous sommes rendu compte que ce problème ne se manifestait que lorsque le DDServer sollicité était branché sur la base réelle (/data/DDBASE) de « manunja ».

Afin de pouvoir débogger DDServer en ajoutant des logs supplémentaires, une autre compilation a été effectuée (cf. §11) sur « manunja » et démarrée sur la base réelle.

Il se trouve qu'avec cette nouvelle compilation, le problème de performance disparaît...

Par ailleurs, un bug de fermeture des fichiers netCDF de la base a été corrigé : la fermeture de certains fichiers pouvait être sollicitée sur des fichiers déjà fermés. Avec la compilation initiale, l'erreur n'était pas relevée, alors qu'avec la nouvelle elle sortait en exception.

Suite à cette correction, et après avoir vérifié longuement les temps d'exécution des appels à la fonction « DD_GetMultiData » par le module AMDA_Kernel, en utilisant la nouvelle compilation de DDServer, nous pouvons aujourd'hui affirmer que le problème de performance ne se produit plus.

En revanche, nous ne pouvons pas expliquer de manière certaine les raisons du dysfonctionnement de l'ancienne compilation (erreur lors de la compilation ? influence du fait que les bibliothèques netCDF et DD soient liées en statique pour cette compilation ? ...).

14 COMPILATION DES PARAMETRES DERIVES AU MOMENT DE LEUR SAUVEGARDE

L'exécutable « amdaParameterGenerator » du module AMDA_Kernel permet d'effectuer la pré-compilation d'un paramètre dérivé.

Le nouveau type de requête « PARAMSGEN » a été ajouté au niveau du « RequestManager » permettant d'appeler cet exécutable sur un paramètre donné.

Ce type de requête est appelé par la fonction « compilParam » du « AmdaAction » du module AMDA_IHM.

L'appel s'effectue de manière silencieuse au moment de la sauvegarde d'un paramètre dérivé par l'utilisateur. En cas d'échec de la compilation, un message d'information prévient l'utilisateur.

A la suppression du nœud d'un paramètre dérivé, la fonction « realDelete » de la classe « InteractiveNode » appelle la fonction « compilParamDelete » du « AmdaAction » afin de supprimer, si elle existe, la librairie compilée de l'espace de travail de l'utilisateur.

En cours de réalisation de cette tâche, un bug a été détecté. Par exemple, une compilation d'une expression simple telle que « imf[0]*3 » échouée.

Une correction a donc été apportée au parser d'expression du module AMDA_Kernel, et un test de validation supplémentaire ajouté dans Fitnessse (ReleaseS.ZatAmda2015.BugParse) pour couvrir ce cas.

15 MODIFICATION DE LA PALETTE DE COULEURS UTILISEE POUR LES PLOTS

Plutôt que de redéfinir globalement la palette de couleurs dans le module AMDA_Kernel, nous avons fait en sorte que le module AMDA_Integration utilise sa propre palette de couleur.

Ainsi, cette nouvelle palette ne s'applique pas lors des passages des tests Fitness. Il n'a donc pas été nécessaire de redéfinir toutes les images de référence des tests.

Pour ce faire, nous avons ajouté le fichier de configuration spécifique au module AMDA_Integration « config/plotConfig.xml » faisant référence à la palette par défaut « config/plplot/cmap0_default.pal ».

Les fichiers de propriétés « app.properties » de AMDA_Kernel, générés par AMDA_Integration, font références à ce nouveau fichier de configuration du plot (clé « app.plot.configfile »).

Enfin, dans ce fichier « config/plplot/cmap0_default.pal », 16 couleurs sont définies de manière à avoir les trois premières couleurs identiques à celles utilisées par l'ancien noyau d'AMDA.

16 AMELIORATION DE L’AFFICHAGE INTERACTIF DES PLOTS

Le module AMDA_Integration utilise dorénavant son propre fichier de configuration pour le plot (cf. §15).

Dans ce fichier « plotConfig.xml », l'attribut « dpi » du nœud « page » permet de définir une résolution propre au module AMDA_Integration.

Après étude des plots réalisés par l'ancien noyau, il a été déterminé que la résolution à appliquer pour le nouveau noyau était de 90dpi.

Afin de préserver les proportions des images réduites par le module AMDA_IHM, les dimensions des images produites par le nouveau noyau ont été corrigées dans le fichier « js/app/controllers/InteractivePlotMgr.js ».

Ces deux modifications ont permis d'améliorer drastiquement l'affichage des plots dans le module AMDA_IHM.

17 ALIGNEMENT DES « PLOTS AREA » AVEC LE « LAYOUT » MANUEL

La fonction « computePanelPlotAreaBoundsForManualLayout » a été ajoutée à la classe « PlotOutput » afin d'aligner les bords gauches des « plots area » des « TimePlot » dont les panels partagent le même bord gauche, et ce pour le cas d'un layout de type « manual ».

De même, la fonction « computePanelLegendPosition » a été modifiée afin d'aligner les légendes dans ces conditions.

Suite à ces modifications, les images de référence des tests suivants ont été modifiées :

- « ThirdPart/ReRelease2/Sprint1/UserStory76 » :
 - Test 29
 - Test 30
 - Test 31
- « ThirdPart/ReRelease2/Sprint2/UserStory82 » :
 - Test 00
- « ThirdPart/ReRelease2/Sprint3/UserStory91 » :
 - Test 00

18 INFORMATION DE CALIBRATION POUR PARAMETRES ISSUS DE FICHIERS LOCAUX

18.1 DEFINITION DES INFORMATIONS DE CALIBRATION AU NIVEAU DU MODULE AMDA KERNEL

18.1.1 Définition d'une information de calibration liée au ParamGet LocalFileInterface

A la manière de ce qui a été mis en place pour le ParamGet DDServerInterface, le nœud « clb » a été introduit au niveau du nœud « param » d'un paramètre local (cf. « config/DataBaseParameters/xsd/parameter/getlocalfile.xsd »).

Ce nœud contient un attribut « name » définissant la clé désignant l'information de calibration que le ParamGet doit récupérer.

18.1.2 Définition d'une information de calibration de manière manuelle (non liée à un ParamGet)

Une information de calibration peut être définie manuellement dans un fichier de définition d'un paramètre, en définissant un nœud « clbManual » au niveau du nœud « param » (cf. « config/DataBaseParameters/xsd/parameter/parameter.xsd »).

Ce nœud peut contenir un ensemble de nœuds « value » définissant la liste des valeurs de l'information de calibration.

18.2 PRISE EN COMPTE DES INFORMATIONS DE CALIBRATION AU NIVEAU DU MODULE AMDA KERNEL

18.2.1 Prise en compte d'une information de calibration liée au ParamGet LocalFileInterface

La classe abstraite « FileReaderAbstract » du « LocalFileInterface » intègre la nouvelle fonction virtuelle « getInfo » dont la responsabilité est de renvoyer une liste des valeurs d'une information de calibration.

Dans le cas :

- De l'implémentation « FileReaderVOTable », cette fonction cherche à lire les informations de calibration dans les nœuds « PARAM » du fichier VOTable,
- De l'implémentation « FileReaderASCII », cette fonction cherche parmi les lignes commentées le mot clé « CALIB_INFO » définissant une ligne contenant une information de calibration. Ce qui suit ce mot clé est « splitté » en utilisant le séparateur « , ». La première valeur correspond au nom de l'information de calibration, alors que les suivantes correspondent aux valeurs de calibration (cf. fichier d'exemple « config/LocalBase/cassini_CAPS_01.txt »).
- De l'implémentation « FileReaderCDF », cette fonction ne produit rien pour le moment. En effet, les informations de calibration pour ce type de fichier sont généralement contenues dans une variable, hors

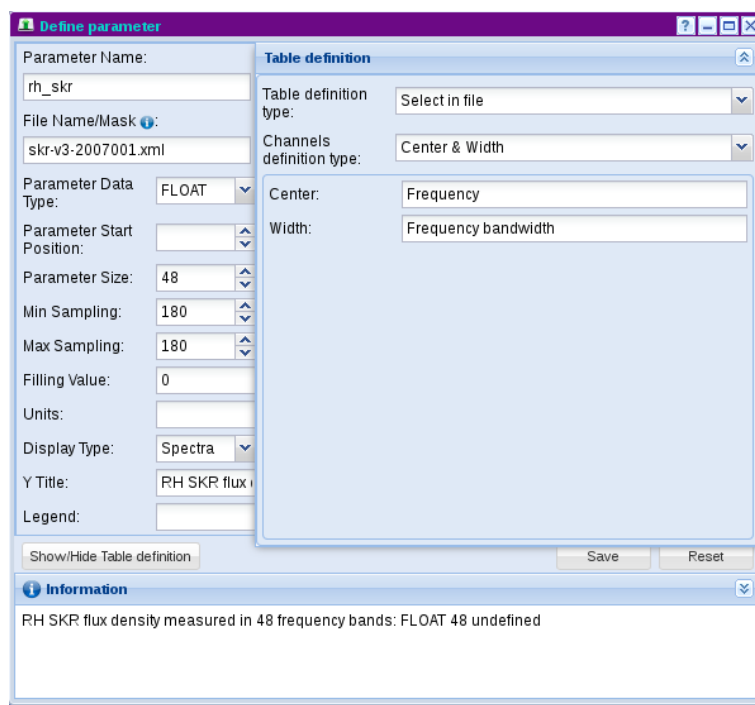
seul le cas d'une information de calibration constante au cours du temps est implémenté au niveau du module AMDA_Kernel.

18.2.2 Prise en compte d'une information de calibration « manuelle »

Les informations de calibration « manuelle » sont lues depuis la classe « ParamNode » du « XMLConfigurator ».

18.3 DEFINITION 'UNE TABLE AU NIVEAU DU MODULE AMDA_IHM

Depuis l'interface « Define parameter » permettant de définir un paramètre depuis un fichier uploadé par un utilisateur, le bouton « Show/Hide Table definition » permet d'afficher, ou de cacher, l'interface de définition d'une table associée au paramètre :

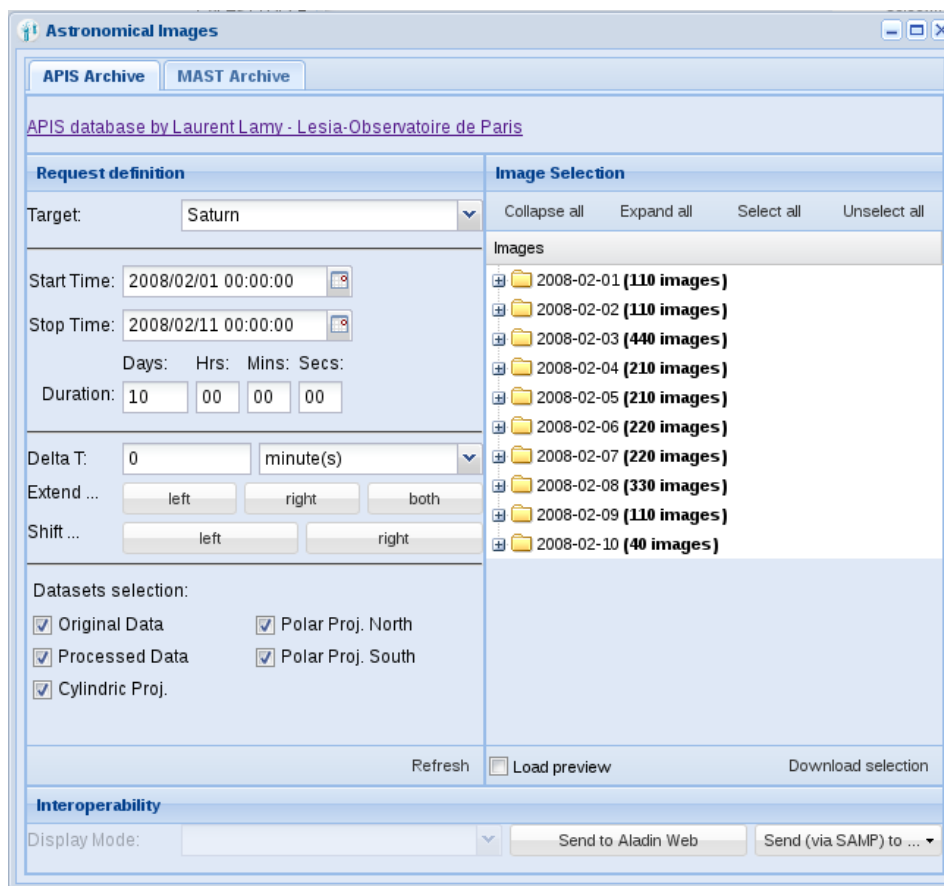


La table ainsi définie est sauvegardée dans le fichier XML de définition du paramètre dans l'espace de travail de l'utilisateur.

19 PRISE EN COMPTE DES MODIFICATIONS DU SERVICE DE LA BASE APIS DANS AMDA

Suite à une modification du service EPN-TAP de la base APIS, l'interface de visualisation de ces images n'était plus adaptée dans AMDA.

Les images sont notamment regroupées sous forme de « datasets » par le service. L'utilisateur peut dorénavant sélectionner les datasets sur lesquels il veut réaliser sa requête :



La liste des « targets » a également été enrichie avec l'apparition des lunes : Europa, Ganymede, Io et Titan.

Enfin, des corrections mineures ont été apportées afin de prendre en compte les modifications du service et construire correctement l'arbre associé dans AMDA.

20 PRISE EN COMPTE DES FORMATS DE TEMPS EPOCH16 ET TT2000 DANS L'APPLICATION « CONVERT2NC »

La gestion des temps de type « Epoch16 » et « TT2000 » a été ajoutée dans l'application « convert2nc ». Les modifications ont été committées dans le repository SVN « depotDECODER ».

Il est à noter que cette application convertit des fichiers dans un format netCDF adapté à DDServer. Ainsi le temps est converti au format « DD Time », ainsi la précision du temps s'arrête à la milliseconde.

21 MODIFICATIONS AUTOUR DE L'APPLICATION TREPS

Des modifications ont été apportées à l'application TREPS.

Un repository GIT a été créé pour cette application : <https://gitlab.irap.omp.eu/CDPP/TREPS.git>

De plus, l'application est passée de la version 1.7.0 à la version 1.8.0.

21.1 RAPPEL SUR LA MAUVAISE GESTION DES NAN PAR LE WEBSERVICE 3DVIEW

L'IRAP avait détecté un fichier source de données qui provoquait une exception de type « Error during the call of listNewFrameAtt of 3DView Web Service (For input string : « y ») ».

Après vérification, l'erreur est due à la présence de NaN dans les données fournies en entrées. TREPS gère correctement ces valeurs, en revanche c'est le Web Service 3DView qui sort en exception lorsqu'il en reçoit.

La correction doit donc être apportée au niveau de 3DView.

21.2 OUVERTURE DE L'AIDE A L'ENTREE DE L'APPLICATION

La propriété « collapsed » de « ihm/app/view/Application/Help/HelpPanel.js » a été initialisée à « false » afin d'ouvrir le panneau d'aide à l'entrée de l'application.

21.3 AMELIORATION DU MECANISME DE CONVERSION DU CHAMP « TEMPS » D'UN FICHIER SOURCE

TREPS peut être utilisé afin de convertir le champ « temps » d'un fichier dans un autre format.

Afin de simplifier ce cas d'utilisation, les modifications suivantes ont été apportées :

- Il n'est plus nécessaire de sélectionner deux fois le même repère pour réaliser une conversion sans transformation. L'utilisateur peut se contenter de laisser les deux ComboBox vides,
- Lorsqu'aucune transformation n'est appliquée, le ComboBox de définition de la structure d'exportation est grisé. En effet, seule la structure « Same Structure » a un sens dans ce cas de figure,
- Lorsque le champ « temps » est converti, le champ initial du fichier source n'est plus répété dans le fichier exporté.

21.4 PRECISION DES NOMBRES FLOTTANTS DANS LE CAS D'UN FICHIER SOURCE ASCII

Un fichier ASCII n'étant pas auto-décrit, le type des différentes colonnes détectées n'est pas connu. L'application TREPS les considérerait donc tous comme des « floats ».

Rapport des activités lors de l'Assistance Technique AMDA 2015

Cela provoque des erreurs de précision en cas de « valeurs élevées ».

Pour être moins sensible à ce problème, les différentes colonnes sont dorénavant vues comme des « double ».

22 DOCUMENTS APPLICABLES ET DE REFERENCE (A/R)

A/R	Référence	Titre
[R0]	PRC-AMD-1501	Proposition commerciale – Assistance technique pour des développements sur l'application CDPP/AMDA
[R1]	CDPP-MI-32500-440-SI	Manuel d'installation de AMDA Kernel
[R2]	CDPP-MI-32500-463-CS	Noyau AMDA-NG – 2 nd e partie – Manuel d'installation

23 GLOSSAIRE ET ABREVIATIONS

23.1 GLOSSAIRE

Terme	Définition

23.2 ABREVIATIONS

Abréviation	Nom détaillé
IRAP	Institut de Recherche en Astrophysique et Planétologie
CDPP	Centre de Données de la Physique des Plasmas
AMDA	Automated Multi-Dataset Analysis
TREPS	Transformation de REpères en Physique Spaciale
APIS	Auroral Planetary Imaging and Spectroscopy