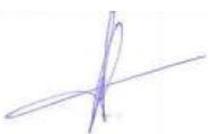


Agence ou Service : NTIC

Projet : Développement du noyau AMDA-NG (3ème partie) et intégration avec l'IHM

NOYAU AMDA-NG (3EME PARTIE) ET INTEGRATION AVEC L'IHM AMDA - PLAN DE TEST

Rédigé par : Benjamin Renard Mathias Mazel	Diffusé à : CNES / IRAP 
Approuvé par : Chef de projet AKKA – N. Lormant Responsable projet CNES – N. Dufourg	

LISTE DES MODIFICATIONS DU DOCUMENT

Vers.	Date	Paragraphes modifiés	Description des modifications
01.0	31/10/14	Tout	Création du document

SOMMAIRE

1	INTRODUCTION	6
1.1	Objet du document	6
1.2	Domaine d'application	6
2	STRATEGIE DE TESTS	8
2.1	Objectifs/Limites	8
2.2	Principes généraux.....	8
2.3	Eléments concernés.....	9
2.4	Responsabilités	9
2.5	Documents utilisés	10
2.6	Règles d'identification	10
2.6.1	Identification des tests	10
2.6.2	Identification des résultats de tests.....	11
2.7	Environnement de tests.....	12
2.7.1	Configuration matérielle	12
2.7.2	Outils	13
2.8	Critères d'arrêt des tests	14
2.9	Planning et charges.....	14
2.10	Déroulement de l'activité de tests.....	15
2.11	Gestion des anomalies	15
3	TESTS UNITAIRES.....	16
3.1	Objectifs/limites	16
3.2	Documents de référence	16
3.3	Eléments concernés.....	16
3.4	Jeux de données	16
3.5	Méthodologie de tests	16
4	TEST D'INTEGRATION	17
4.1	Objectifs/Limites	17
4.2	Documents de référence	17
4.3	Eléments concernés.....	17
4.4	Jeux de données	17
4.5	Méthodologie.....	17

4.6	Scénarios de tests	18
4.7	Résultats de tests	18
5	TESTS DE VALIDATION	20
5.1	Objectifs/Limites	20
5.2	Documents de référence	20
5.3	Éléments concernés	20
5.4	Jeux de données	20
5.5	Méthodologie	20
5.6	Scénarios de tests	20
5.6.1	Cas test module AMDA_Integration: Download – Résultat immédiat – US42	21
5.6.2	Cas test module AMDA_Integration: Data Mining – Résultat immédiat – US44	26
5.6.3	Cas test module AMDA_Integration: Batch Mode – US45	29
5.6.4	Cas test module AMDA_Integration: Plot – US43	31
5.6.5	Cas test module AMDA_Integration : Paramètre dérivé – US78.....	35
5.7	Résultats de tests	37
6	TESTS DE NON-REGRESSION.....	38
6.1	Methodologie de tests de non-régression	38
6.2	Scénarios de tests	38
6.3	Résultats de tests	38
7	MATRICE DE COUVERTURE DES TESTS.....	39
8	CAHIER DE RECETTE	40
8.1	Domaine d'application	40
8.2	Rappel des fonctionnalités du produit	40
8.3	Architecture du produit	40
8.4	Couverture de la recette	40
8.5	Fournitures à recetter	41
8.5.1	Les logiciels.....	41
8.5.2	Les documents.....	41
8.5.3	Les matériels.....	41
8.6	Contexte de la recette	41
8.7	Les essais de recette	42
8.7.1	Installation – Paramétrage	42
8.7.2	Les tests fonctionnels.....	42
8.7.3	Les contraintes.....	42
8.8	Logique d'enchaînement des essais	42

8.8.1	Dossier de référence.....	42
8.8.2	Liste des tests	42
8.9	Organisation de la recette	42
8.10	Planning.....	43
8.11	Intervenants.....	44
8.12	Résultats de la recette.....	44
8.12.1	Conditions d'acceptation.....	44
8.12.2	Gestion des anomalies détectées en recette.....	44
9	ANNEXE.....	45
9.1	Analyse du rapport Sonar.....	45
9.1.1	Commentaires.....	45
9.1.2	Duplications.....	45
9.1.3	Violations majeures.....	45
9.1.4	Violations mineures.....	46
9.1.5	Couverture du code par les tests	46
9.2	Fiches de tests	46
9.2.1	Cas test module AMDA_Integration: Download – Résultat immédiat – US42	46
9.2.2	Cas test module AMDA_Integration: Data Mining – Résultat immédiat – US44	51
9.2.3	Cas test module AMDA_Integration: Batch Mode – US45	54
9.2.4	Cas test module AMDA_Integration: Plot – US43	56
9.2.5	Cas test module AMDA_Integration : Paramètre dérivé – US78.....	60
10	DOCUMENTS APPLICABLES ET DE REFERENCE (A/R).....	63
11	GLOSSAIRE ET ABREVIATIONS.....	64
11.1	Glossaire.....	64
11.2	Abréviations	65

1 INTRODUCTION

1.1 OBJET DU DOCUMENT

L'objectif de l'activité de tests est de vérifier et démontrer la conformité du logiciel aux spécifications validées.

Le présent document décrit l'activité de tests mise en œuvre par l'équipe projet AKKA Technologies dans le cadre du projet « Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA ».

Il est constitué par le plan de tests, qui décrit la stratégie de tests, le déroulement global de l'activité de tests, ainsi que son suivi.

Le document est complété au fur et à mesure de l'avancement du projet.

Dans le cadre de ce projet développé en mode agile (SCRUM), nous entendons par spécifications validées les Users Stories (USs) prises en comptes et leurs Tests d'Acceptations (TAs).

Les USs et les TAs sont de la responsabilité du Product Owner.

Ce document fait également office de cahier de recette (cf. §8). Il décrit les modalités de déroulement de la phase de recette, en précisant notamment les prérequis, le périmètre, les moyens mis en œuvre, les conditions de passage des tests, les critères d'acceptation et le planning.

1.2 DOMAINE D'APPLICATION

Ce document couvre la stratégie de test appliquée :

- Aux évolutions et corrections apportées au module « AMDA_Kernel »,
- Au développement du module « AMDA_Integration ».

Module « AMDA Kernel » :

Le module « AMDA_Kernel » est constitué d'un ensemble de bibliothèques communes utilisées par quatre programmes s'exécutant en ligne de commande :

- « EXE_AMDA_Kernel » : instancié lors de la première phase du projet, cet exécutable est utilisé pour produire un fichier de sortie au format ASCII Tabulaire suivant les arguments de la ligne de commande, à partir de données de paramètres externes (par exemple des données de DDServer),
- « amdaXMLRequestorTool » : instancié lors de la première phase du projet, cet exécutable est utilisé pour produire une sortie à partir d'une requête écrite dans un fichier XML et fournie en ligne de commande, à partir de données de paramètres externes (par exemple des données de DDServer),
- « amdaParameterGenerator » : instancié lors de la première phase du projet, cet exécutable est utilisé pour produire le code relatif à un paramètre décrit dans un fichier XML et fournie en ligne de commande,
- « ttOperations » : instancié lors de la deuxième phase du projet, cet exécutable est utilisé pour réaliser des opérations sur des TimeTable.

Pour être testé, le module « AMDA_Kernel » doit impérativement être connecté au serveur DDServer (machine : manunja.cesr.fr, port : 5000) pour fonctionner. L'hébergement et le maintien du serveur de test DDServer est de la responsabilité de l'IRAP.

Le comportement des évolutions et corrections apportées à ce module « AMDA_Kernel » durant ce projet est entièrement spécifié sous forme de USs et Tas dans l'outil WEB IceScrum (Outil de gestion de projet en mode Scrum : <http://bas-agiletools-01.akka.eu:8080/icescrum/p/AKKA3#project>).

Seuls les USs et TAs déjà réalisés ou en cours de réalisation peuvent être considérés comme valides et à tester.

Exemple, si des tests de performance sont à faire, ils doivent être écrits sous forme de US et de TA, de même pour les tests de robustesse. De manière générale, tout test répond à au moins une US.

Module « AMDA_Integration » :

Le module « AMDA_Integration » est constitué d'un ensemble de scripts PHP constituant une interface d'accès au module « AMDA_Kernel ».

Seul le code relatif à l'interface avec le client « AMDA_IHM » a été développé et est concerné par les tests. Le client « AMDA_IHM » est utilisé comme un COTS par le module « AMDA_Integration ».

Le comportement de ce module « AMDA_Integration » est entièrement spécifié sous forme de USs dans l'outil WEB IceScrum (Outil de gestion de projet en mode Scrum : <http://bas-agiletools-01.akka.eu:8080/icescrum/p/AKKA3#project>).

Seuls les USs déjà réalisés ou en cours de réalisation peuvent être considérés comme valides et à tester.

Les tests de validation de ce module sont rédigés sous forme de cas tests.

Ce document décrit, pour le module « AMDA_Kernel » et le module « AMDA_Integration » :

- Les tests unitaires
- Les tests d'intégration
- Les tests de validation
- Les tests de non-régression

2 STRATEGIE DE TESTS

2.1 OBJECTIFS/LIMITES

La stratégie des tests a pour objectif de définir le cadre de l'activité des tests en termes de principes généraux, de définition du contenu des tests, d'enchaînement des étapes et de responsabilités.

2.2 PRINCIPES GENERAUX

Quatre types de tests sont effectués sur le projet :

- Les tests unitaires : tests effectués par le développeur en cours de sprint sur les US qu'il développe. Ces tests sont effectués dans l'espace de développement.
- Les tests d'intégration : tests effectués en cours de sprint sur les US courant du sprint dans l'espace d'intégration. Ils comprennent de plus des tests de qualimétrie (vérification de la bonne application des règles de codage, comme le taux de commentaire, la non duplication de code, la bonne gestion de la mémoire, ...)
- Les tests de validation : tests effectués en fin de sprint sur les US définis dans le sprint.
- Les tests de non-régression : tests effectués tous au long du projet sur les US validées dans les sprints précédant.

Dans le cadre de ce projet, nous nous appuyons totalement sur les tests d'acceptation (pour le module « AMDA_Kernel ») et les cas tests définis (pour le module « AMDA_Integration ») par le « ProductOwner ». Ces tests d'acceptation, et cas tests, peuvent être fonctionnels, structurels, de performance, de robustesse etc.

Les mêmes scénarii de tests sont utilisés pour les tests unitaires, d'intégration, de validation et de non régression.

Nous suivons quelques principes de la méthode «BehaviourDevelopmentDriven» BDD:

- La « User Story » et ses tests d'acceptation forment une spécification qui est travaillée par l'équipe SCRUM afin d'obtenir un langage naturel commun entre les développeurs et le métier.
- Un développeur, travaillant sur sa plateforme de développement, choisit une US et commence par automatiser les tests d'acceptation qui y sont associés.
- Le développeur les joue pour confirmer leurs échecs.
- Le développeur effectue un premier développement jusqu'à faire passer avec succès ces tests et ceux mis en place pour les précédentes « User Stories ».
- Cette première version est mise en gestion de configuration.
- Le développeur re-factorise enfin le code pour en obtenir une version plus homogène, sans doublon de code et avec un niveau de qualité de code acceptable (Taux de commentaire, pas de fuite et/ou d'écrasement mémoire).
- Cette nouvelle version est mise en gestion de configuration.
- Le Product Owner valide que la « User Story » répond bien aux tests d'acceptation définis lors de la démonstration de fin de Sprint et qu'elle n'a pas d'impact sur les autres US (ou alors de manière acceptable, voir normale).

Le développeur doit rejouer avec succès tous les tests automatisés, ainsi que les cas test, sur la plateforme d'exploitation à chaque livraison de release.

2.3 ELEMENTS CONCERNES

Chaque fonctionnalité décrite par une User Story est testée.

La liste des fonctionnalités se trouve dans l'outil de gestion de projet Scrum:

- IceScrum (<http://bas-agiletools-01.akka.eu:8080/icescrum/p/AKKA3#project>)

Les groupes de fonctionnalités identifiés par le Product Owner se retrouvent sous forme de « features » Scrum:

- Module Plot l'IHM (concerne le module « AMDA_Kernel »),
- Module Data Mining (concerne le module « AMDA_Kernel »),
- Module Statistique l'IHM (concerne le module « AMDA_Kernel »),
- Module « Données utilisateur » l'IHM (concerne le module « AMDA_Kernel »),
- Intégration noyau dans l'IHM (concerne le module « AMDA_Integration »),
- Module « File Output » (concerne le module « AMDA_Kernel »),
- Module « Layout » (concerne le module « AMDA_Kernel »),
- Module « PostProcessing » (concerne le module « AMDA_Kernel »).

2.4 RESPONSABILITES

Le **Product Owner** du projet est responsable de la définition et de la validité des tests d'acceptation, ou des cas tests, des US. Ils doivent être saisis dans IceScrum.

En ce qui concerne le module « AMDA_Kernel », étant donné que les exécutables sont testés en ligne de commande, la majorité des tests sont sous la forme suivante, inspirée du principe « Given Then When »:

- **Arguments Z en entrée du programme X donnera les résultats Y.**

Le Product Owner fournit les entrées (arguments, fichiers d'entrée) et les sorties (fichiers) au travers de IceScrum et FTP.

Le jeu de données DDServer est également de sa responsabilité.

Par exemple, les résultats attendus peuvent être :

- Le code retour du programme,
- Des phrases à trouver dans le fichier de journalisation,
- Les fichiers résultats du traitement,
- ...

La rédaction des cas test pour le module « AMDA_Integration » est également de la responsabilité du Product Owner.

L'équipe de développement SCRUM AKKA est responsable de l'automatisation de ces tests pour le module « AMDA_Kernel ».

Elle est également responsable du passage avec succès de ces tests, ainsi que des cas tests définis.

Une « User Story » ne peut être présentée que lorsque ses tests d'acceptation, ou cas test, passent ainsi que tous les tests des User Stories précédemment déclarées terminées.

Le Product Owner est l'arbitre des tests que l'équipe présente comme obsolètes. En effet, une US peut voir sa fonctionnalité surchargée, contredite, ... par une nouvelle US.

2.5 DOCUMENTS UTILISES

Les scénarii des tests sont des pages FitNesse (pour le module « AMDA_Kernel ») ou des cas tests rédigés (pour le module « AMDA_Integration »).

Les pages FitNesse sont gérées en configuration sous SVN (<http://svn-06.akka.eu/svn/AMDA-KERNEL/>) dans le même projet que les sources (répertoire AMDA_Kernel/test/FitNesseRoot).

Les rapports des tests sont générés et gérés dans Jenkins, Sonar et FitNesse.

Voir le §2.7.2 pour plus d'information sur ces outils.

2.6 REGLES D'IDENTIFICATION

2.6.1 Identification des tests

Pour le module AMDA_Kernel, les tests d'acceptation sont identifiés via les numéros fournis par IceScrum :

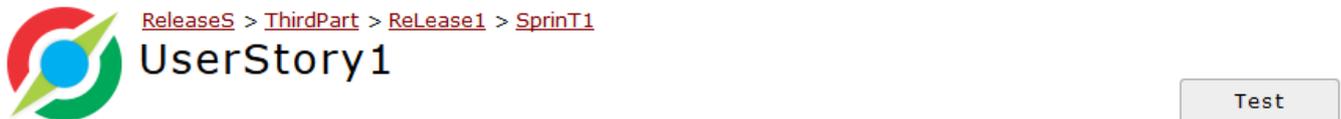
- Numéro de la US
- Numéro du test d'acceptation

Exemple : US 32: Id 31

Sous FitNesse une page de test est écrite pour chaque User Story. Cette page contient les tests d'acceptation concernant la US.

Cette page est rangée dans une arborescence XPart/ReleaseY/SprintZ (voir le répertoire d'une livraison test/FitNesseRoot/ReleaseS/).

Chaque page contient un rappel de l'objet de la User Story.



User Story 1: Tracer un label sur un plot

*variable defined: path=../test/FitNesseRoot/ReleaseS/ThirdPart/ReLease1/SprinT1/UserStory1/
variable defined: workingPath=../PlotFitnesseResults/US1/
variable defined: executable=amdaXMLRequestorTool
variable defined: dataBaseParamPath=../config/DataBaseParameters/*

Description du test

Le but du test est de valider le tracé de label en X ou Y en faisant varier couleur, type et taille de police et orientation

Test

Connexion au serveur

script	ConnectToDDServer
check	login 1

script	TestAmdaCommandLine
set	../test/FitNesseRoot/ReleaseS/ThirdPart/ReLease1/SprinT1/UserStory1/ path

Figure 1 - Aperçu d'une page de test Fitnessse d'une US

Pour le module « **AMDA_Integration** », l'identification suivante est utilisée pour les cas test :

<Type de test>-<US>-<Cas> où :

- <Type de test> : TU, TI, TV, NR respectivement pour : Test Unitaire, Test d'Intégration, Test de Validation, Non Régression.
- <US> : identification de la User Story concernée.
- <Cas> : code du cas de test.

2.6.2 Identification des résultats de tests

Pour le module « **AMDA_Kernel** », nous utilisons le langage de FitNesse pour la lecture des résultats de tests.

Les résultats des tests sont classés dans les catégories de conformité suivantes :

- « right » : résultat conforme aux exigences
- « wrong » : résultat non conforme aux exigences
- « exceptions » : test non desservie par le serveur de test.



[ReleaseS](#) > [ThirdPart](#) > [ReLease1](#) > [SprinT1](#)

UserStory1

Output Captured

Test

✓ Test Pages: 1 right, 0 wrong, 0 ignored, 0 exceptions Assertions: 7 right, 0 wrong, 0 ignored, 0 exceptions (9.055 seconds)

Test System: slim:build/bin/CSlimTestServer

User Story 1: Tracer un label sur un plot

*variable defined: path=../test/FitNesseRoot/ReleaseS/ThirdPart/ReLease1/SprinT1/UserStory1/
variable defined: workingPath=../PlotFitnesseResults/US1/
variable defined: executable=amdaXMLRequestorTool
variable defined: dataBaseParamPath=../config/DataBaseParameters/*

Description du test

Le but du test est de valider le tracé de label en X ou Y en faisant varier couleur, type et taille de police et orientation

Test

Figure 2 - Résultat d'exécution d'une page Fitnessse

Pour le module « **AMDA_Integration** », les résultats des tests sont classés, après leur obtention, dans les catégories de conformité suivantes :

- « OK » : résultat conforme aux exigences
- « POK » : résultat faisant l'objet d'une limite donc Partiellement conforme
- « NOK » : résultat non conforme aux exigences

2.7 ENVIRONNEMENT DE TESTS

2.7.1 Configuration matérielle

Infrastructure AMDA

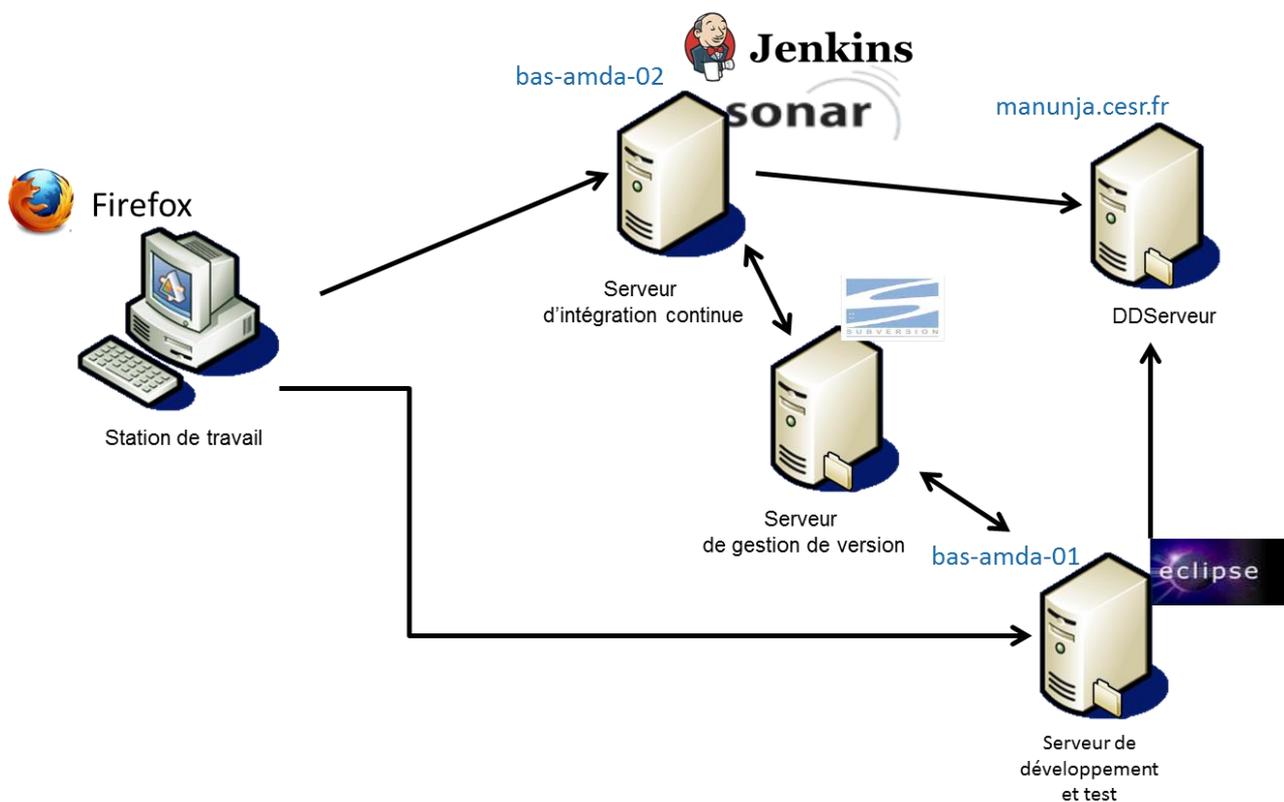


Figure 3 - Infrastructure AMDA

Matériel	Type de machine	Système d'exploitation	Configuration
bas-amda-01	PC processeurs pentium ou équivalents	Centos 6.3	
bas-amda-02			

2.7.2 Outils

2.7.2.1 FitNesse

FitNesse est un outil de développement collaboratif sous forme de WIKI axé sur les tests d'acceptation.

Dans cet outil, l'équipe Scrum:

- Définit les tests d'acceptation sous forme de scénarii,
- Lance l'exécution de tous ou d'une partie de ces tests.

Le principe est que le/les scénario(s) sont passés via le protocole SLIM à un serveur de test qui les exécute et retourne les résultats. FitNesse est donc ordonnanceur et reporteur des tests. Dans notre contexte le serveur de test est codé en C++.

2.7.2.2 Sonar

Sonar est un outil en ligne qui compile différents rapports générés par d'autres outils de validation de code :

- Validation statique de code :
 - o Cppcheck
 - o Cxx-Sonar-plugin
- Validation dynamique de code
 - o Valgrind

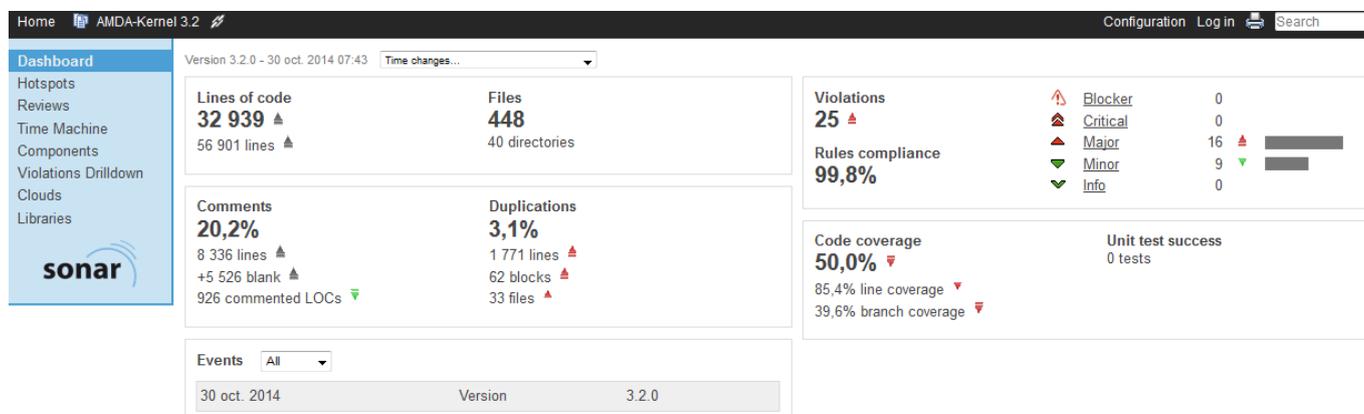


Figure 4 - Compilation des rapports dans Sonar

2.7.2.3 Jenkins

Jenkins est un ordonnanceur de tâche, il permet de faire de l'intégration continue et de lancer FitNesse et les analyseurs de code précédents tous les soirs.

2.7.2.4 Liste des outils utilisés

Outil	Plate-forme	Description	Site
Jenkins	bas-amda-02	open-source continuous integration server	http://jenkins-ci.org/

FitNesse	bas-amda-01 bas-amda-02	scheduler and reporter of tests	http://www.fitnesse.org/
Sonar	bas-amda-02	open source quality management platform	http://www.sonarqube.org/
CXX SONAR Plugin	bas-amda-02	adds basic support of C++ language into Sonar	https://github.com/wenns/sonar-cxx
Cppcheck	bas-amda-01 bas-amda-02	Detects a wide range of problems ranging from performance issues and resource leakage to undefined behaviour. Binary packets are available on/for various platforms. Using the latest release pays off in general; compile from source if in doubt.	http://cppcheck.sourceforge.net/
Valgrind (memcheck)	bas-amda-01 bas-amda-02	Detects various memory management problems at runtime.	http://valgrind.org/

2.8 CRITERES D'ARRET DES TESTS

Test Unitaire

- FitNesse : tous les tests d'acceptation à OK

Test d'intégration :

- FitNesse : tous les tests d'acceptation et tous les cas test à OK
- SONAR :
 - o Valgrind aucune fuite mémoire.
 - o Taux de commentaire > 20%
 - o Pas de duplication de code
 - o Pas de violation de règle majeure.

Test de validation :

- FitNesse : tous les tests d'acceptation et tous les cas test à OK et couvrent toutes les USs livrées

Test de Non régression

- FitNesse : tous les tests d'acceptation et tous les cas test à OK

2.9 PLANNING ET CHARGES

Phase de développement d'une User Story : le temps d'écriture et d'exécution des tests est inclus dans l'évaluation en points de la User Story.

Phase d'intégration : l'automatisation des tests lors de la phase de développement et le passage de ces tests tous les soirs par un outil d'intégration continue, comme Jenkins, réduit le cout des passages de test aux corrections d'éventuelles régressions.

2.10 DEROULEMENT DE L'ACTIVITE DE TESTS

Le déroulement des tests unitaires est à la discrétion du développeur.

Le déroulement des tests d'intégration, de non-régression a lieu de façon automatisé par jenkins toutes les nuits. Le rapport est consultable en ligne sur l'outil tous les matins.

Les tests de validation (via Fitness) et les cas tests sont déroulés pendant la phase de démo de fin de sprint.

2.11 GESTION DES ANOMALIES

L'intégration étant faite en continue, une anomalie constatée, l'a été au plus tôt et son traitement devient la priorité de la journée.

3 TESTS UNITAIRES

3.1 OBJECTIFS/LIMITES

Les tests ont pour objectif de vérifier que le code de la US en développement est OK.

Ces tests sont à la discrétion des développeurs.

Pour le module « AMDA_Kernel », ils se basent sur l'outil FitNesse. Ces tests deviennent les tests d'intégration en fin de développement de la US.

Pour le module « AMDA_Integration », ils sont écrits sous forme de scripts PHP.

3.2 DOCUMENTS DE REFERENCE

IceScrum

3.3 ELEMENTS CONCERNES

Les exécutables du module « AMDA_Kernel » :

- EXE_AMDA_Kernel,
- amdaXMLRequestTool,
- amdaParameterGenerator.
- ttOperations.

Les sources du module « AMDA_Intégration ».

3.4 JEUX DE DONNEES

Voir : §2.4

Les fichiers sont gérés en configuration.

La base de données est fournie par l'IRAP et est une instance dédiée aux tests sur manunja.cesr.fr:5000.

3.5 METHODOLOGIE DE TESTS

Tests boîte noire :

- un cas de test pour des données en entrée de l'élément de plus haut niveau,

La méthodologie utilisée est dite directe, tous les éléments sont intégrés en une seule fois : pas de bouchon.

4 TEST D'INTEGRATION

4.1 OBJECTIFS/LIMITES

Les tests ont pour objectif de vérifier que le code de l'ensemble des développeurs fonctionne encore, après intégration en gestion de configuration.

Pour le module « AMDA_Kernel », ces tests sont automatisés par Jenkins. Ils ne s'appliquent qu'aux tests du sprint courant :

Ils consistent en :

- une récupération des sources à partir de la gestion de configuration (branche trunk).
- une compilation et une exécution des scénarii de tests enregistrés dans FitNesse.

Pour le module « AMDA_Integration », ces tests sont effectués manuellement afin :

- d'installer le module,
- exécuter les cas tests.

4.2 DOCUMENTS DE REFERENCE

IceScrum constitue la référence en ce qui concerne la définition des USs et des TAs.

Les tests de qualité du code s'appuient notamment sur les recommandations CNES décrites dans [R1].

4.3 ELEMENTS CONCERNES

Les exécutables du module « AMDA_Kernel » :

- EXE_AMDA_Kernel,
- amdaXMLRequestTool,
- amdaParameterGenerator.
- ttOperations.

Les sources du module « AMDA_Intégration ».

4.4 JEUX DE DONNEES

Voir : §2.4

Les fichiers sont gérés en configuration.

La base de données est fournie par l'IRAP et est une instance dédiée aux tests sur manunja.cesr.fr:5000.

4.5 METHODOLOGIE

Tests boîte noire :

- un cas de test pour les données en entrée de l'élément de plus haut niveau,

La méthodologie utilisée est dite directe, tous les éléments sont intégrés en une seule fois : pas de bouchon.

4.6 SCENARIOS DE TESTS

Pour le module « AMDA_Kernel »: les scénarios sont décrits par les jobs Jenkins:

- AMDA_Kernel_Phase3_GCOV,
- AMDA_Kernel_Phase3_SONAR,
- AMDA_Kernel_Phase3_RELEASE.

Pour le module « AMDA_Intégration » : les scénarios sont décrits sous forme de cas test cf. §5.6

4.7 RESULTATS DE TESTS

Pour le module « AMDA_Kernel »: Résultats d'exécution des jobs Jenkins:

- AMDA_Kernel_Phase3_GCOV,
- AMDA_Kernel_Phase3_SONAR,
- AMDA_Kernel_Phase3_RELEASE.

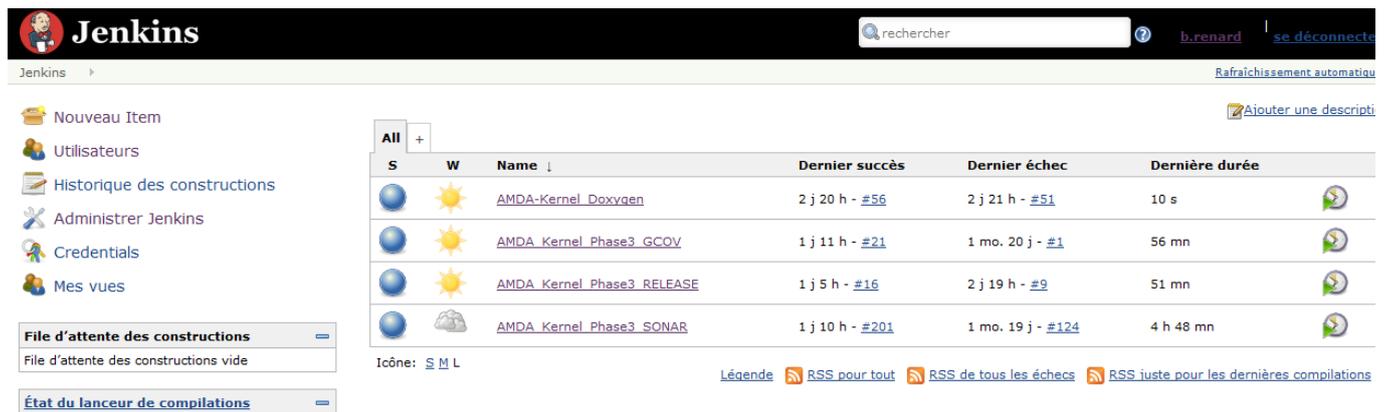


Figure 5 - Résultats d'exécution des jobs Jenkins

Ainsi que la page de rapport SONAR :

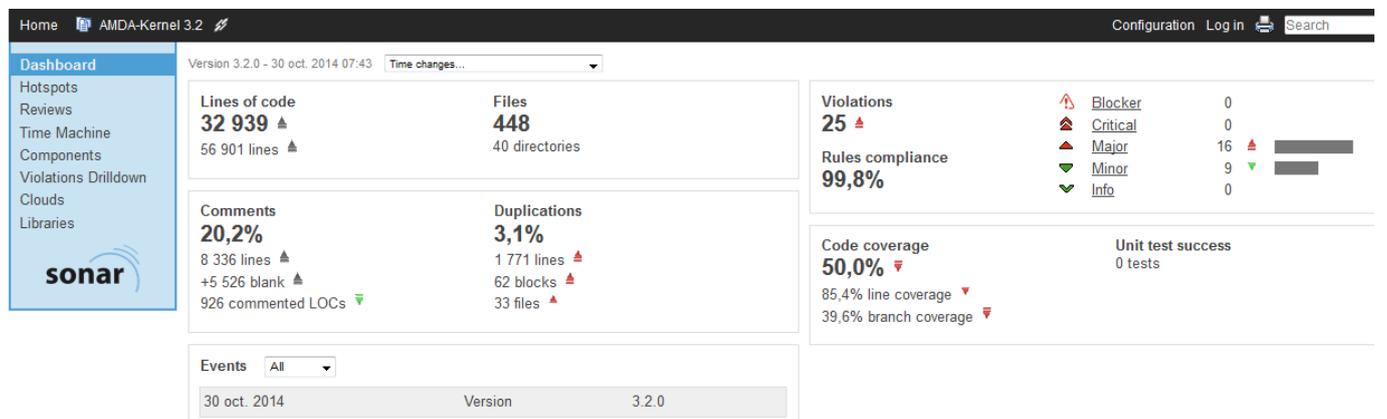


Figure 6 - Rapport SONAR

Une analyse de ce rapport SONAR est proposée en Annexe.

Pour le module « AMDA_Integration » : Résultats d'exécution manuelle des cas test (cf. §5.7).

5 TESTS DE VALIDATION

5.1 OBJECTIFS/LIMITES

Les tests ont pour objectif de vérifier que les objectifs du client sont atteints.

Pour le module « AMDA_Kernel » : Ces tests sont automatisés par Jenkins. Ils sont constitués par l'ensemble des tests de non régression et des tests d'intégration.

Ils consistent en :

- une récupération des sources à partir de la gestion de configuration (branche trunk).
- une compilation et une l'exécution des scénarii de tests enregistrés dans FitNesse.

Pour le module « AMDA_Integration » : Les cas tests doivent être exécutés manuellement.

5.2 DOCUMENTS DE REFERENCE

IceScrum

5.3 ELEMENTS CONCERNES

Les exécutables du module « AMDA_Kernel » :

- EXE_AMDA_Kernel,
- amdaXMLRequestTool,
- amdaParameterGenerator.
- ttOperations.

Les sources du module « AMDA_Intégration ».

5.4 JEUX DE DONNEES

Voir : §2.4

Les fichiers sont gérés en configuration.

La base de données est fournie par l'IRAP et est une instance dédiée aux tests sur manunja.cesr.fr:5000.

5.5 METHODOLOGIE

Tests boîte noire :

- un cas de test pour les données en entrée de l'élément de plus haut niveau,

La méthodologie utilisée est dite directe, tous les éléments sont intégrés en une seule fois pas de bouchon.

5.6 SCENARIOS DE TESTS

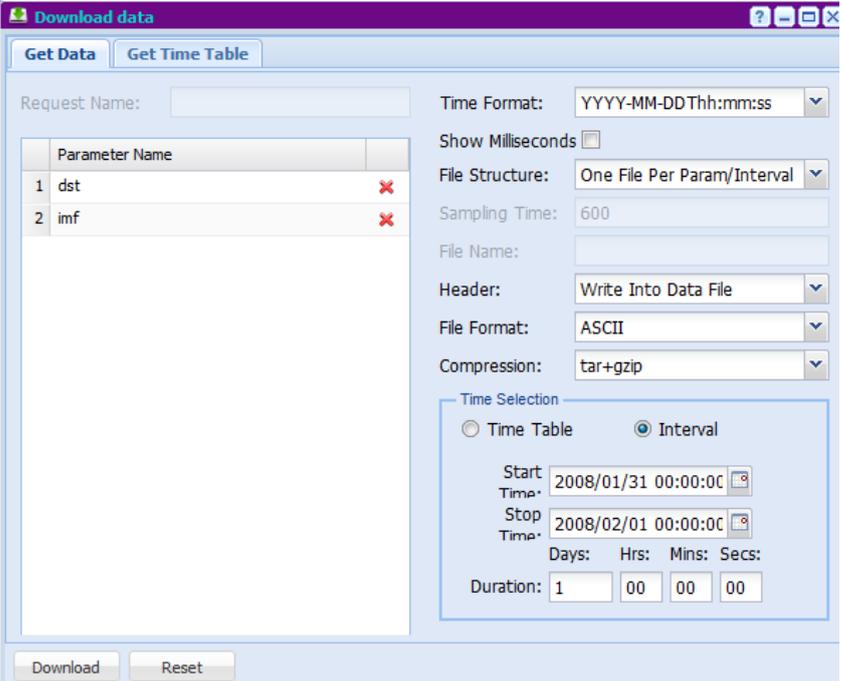
Pour le module « AMDA_Kernel » : les scénarios sont décrits par les jobs Jenkins:

- AMDA_Kernel_Phase3_GCOV,
- AMDA_Kernel_Phase3_SONAR,

AMDA_Kernel_Phase3_RELEASE.

Pour le module « AMDA_Integration », les scénarios présentés dans la suite de cette section constituent les cas test du module :

5.6.1 Cas test module AMDA_Integration: Download – Résultat immédiat – US42

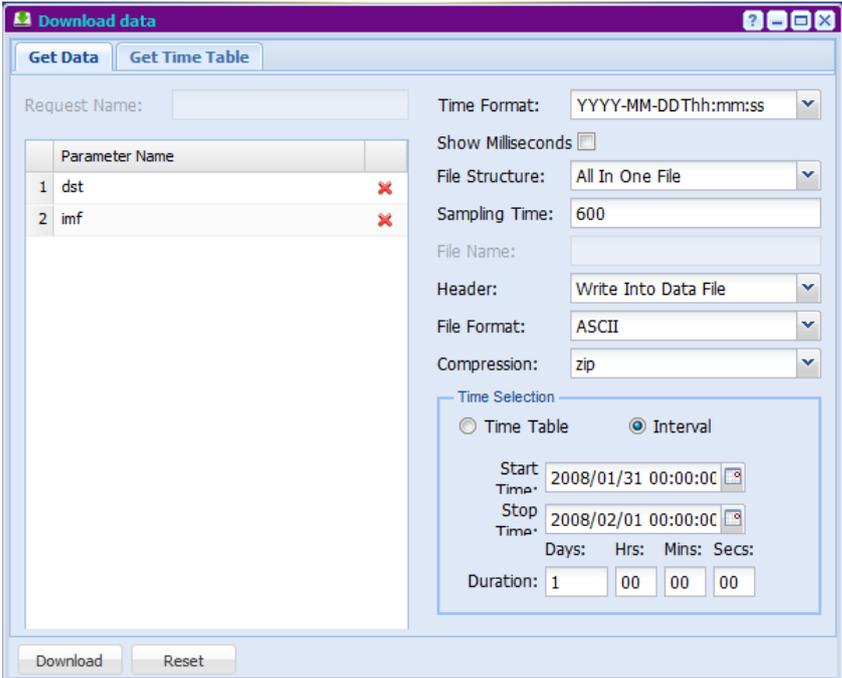
Titre du test	Download – Résultat immédiat – Deux paramètres – Structure « One File Per Param/Interval » - Compression « tar+gzip »	Numéro de Test	TV-US42-001	
Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »	OK	Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.	OK	Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.
4	Observez l'archive produite	L'archive doit porter l'extension « tar.gz ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir deux fichiers : « output-dst_2008030000000000.txt » « output-imf_2008030000000000.txt »	OK	
5	Editez le fichier « output-dst_2008030000000000.txt »	Le fichier s'ouvre et contient un header et des données. Le temps d'échantillonnage est 3600s.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
6	Editez le fichier « output-imf_2008030000000000.txt »	Le fichier s'ouvre et contient un header et des données. Le temps d'échantillonnage est 16s.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

Titre du test	Download – Résultat immédiat – Deux paramètres – Structure « All In One File » - Sampling time « 600 » - Compression « zip »	Numéro de Test	TV-US42-002
----------------------	---	-----------------------	--------------------

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

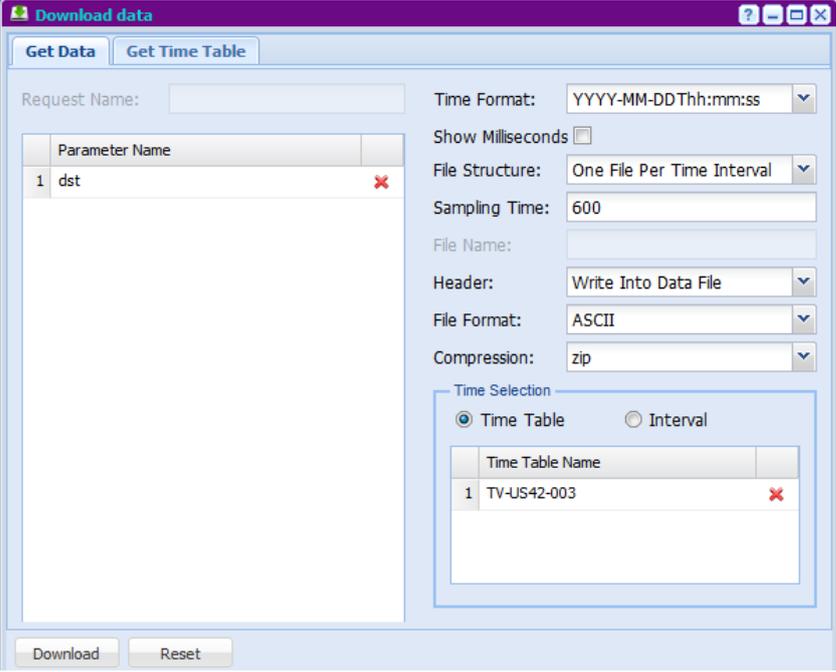
Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »	OK	Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.	OK	Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

4	Observez l'archive produite	L'archive doit porter l'extension « zip ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir un seul fichier : « output-dst_imf_2008030000000000.t xt »	OK	
5	Editez le fichier « output-dst_imf_2008030000000000.t xt »	Le fichier s'ouvre et contient un header et des données. Le temps d'échantillonnage est 600s.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

Titre du test	Download – Résultat immédiat – Un paramètre – Structure « One File Per Time Interval » - Sampling time « 600 » - Compression « zip »	Numéro de Test	TV-US42-003
----------------------	---	-----------------------	--------------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition	<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>TimeTable créée portant le nom « TV-US42-003 » et contenant les intervalles de temps : « 2008-01-31T00:00:00 - 2008-01-31T04:00:00 » et « 2008-01-31T08:00:00 - 2008-01-31T12:00:00 »</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 			
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »	OK	Si le résultat n'est pas immédiat et passe par un job, veuillez réduire les intervalles de temps de la time table données en requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.	OK	Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.

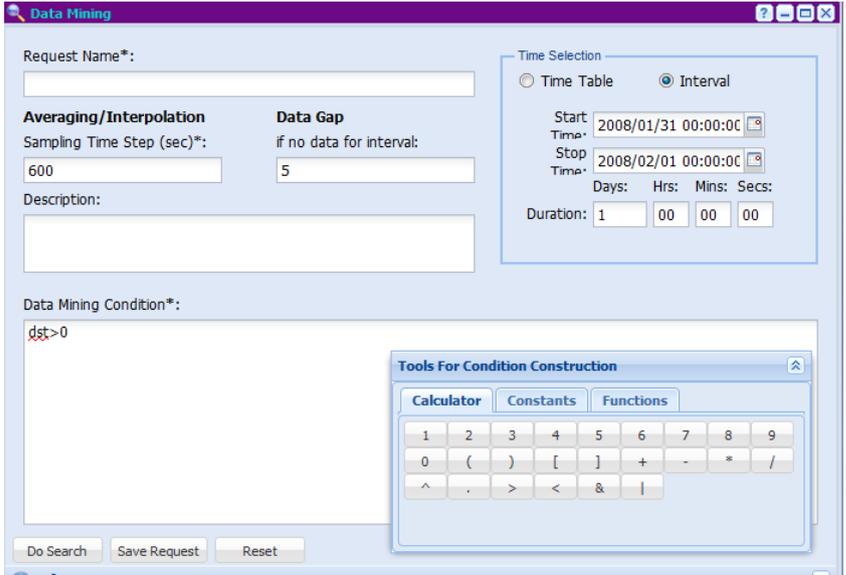
**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

4	Observez l'archive produite	L'archive doit porter l'extension « zip ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir trois fichiers : « output-dst_0.txt », « output-dst_1.txt » et « output-dst_info.txt »	OK	
5	Editez le fichier « output-dst_info.txt »	Le fichier s'ouvre et contient uniquement le header.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
6	Editez le fichier « output-dst_0.txt »	Le fichier s'ouvre et contient les données relatives au premier intervalle de temps.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
7	Editez le fichier « output-dst_1.txt »	Le fichier s'ouvre et contient les données relatives au deuxième intervalle de temps.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

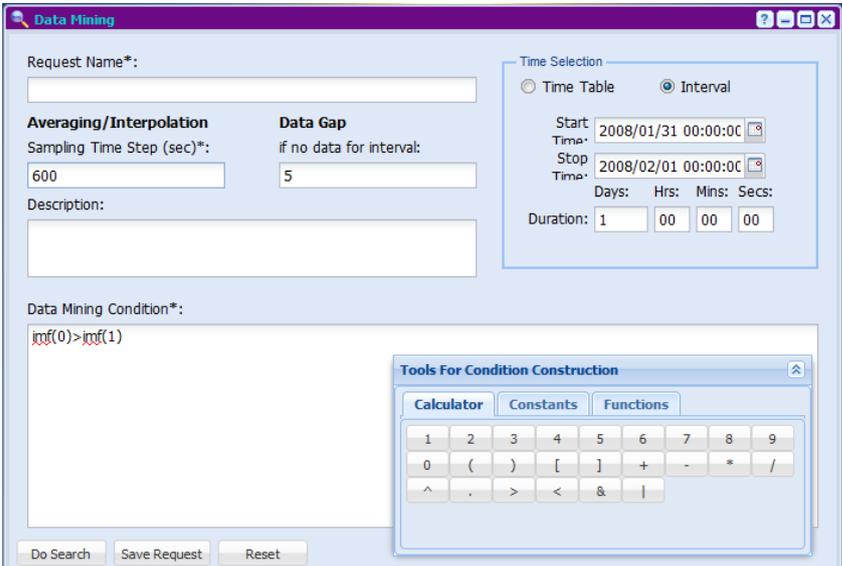
5.6.2 Cas test module AMDA_Integration: Data Mining – Résultat immédiat – US44

Titre du test	Data Mining – Résultat immédiat – Condition « dst > 0 »	Numéro de Test	TV-US44-001
----------------------	---	-----------------------	--------------------

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Data Mining ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Do Search »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Attendez la fin d'exécution de la requête	A l'issu de l'exécution, la fenêtre « Results » s'affiche avec un item « datamining_XXXX »	OK	Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Edit/Save » pour la TimeTable de l'item « Results ».	L'interface d'édition d'une TimeTable s'ouvre avec les données de la TimeTable calculée	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
4	Cliquez sur le bouton « Edit/Save » pour la Gaps TimeTable de l'item « Results ».	L'interface d'édition d'une Gaps TimeTable s'ouvre avec les données de la TimeTable calculée	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

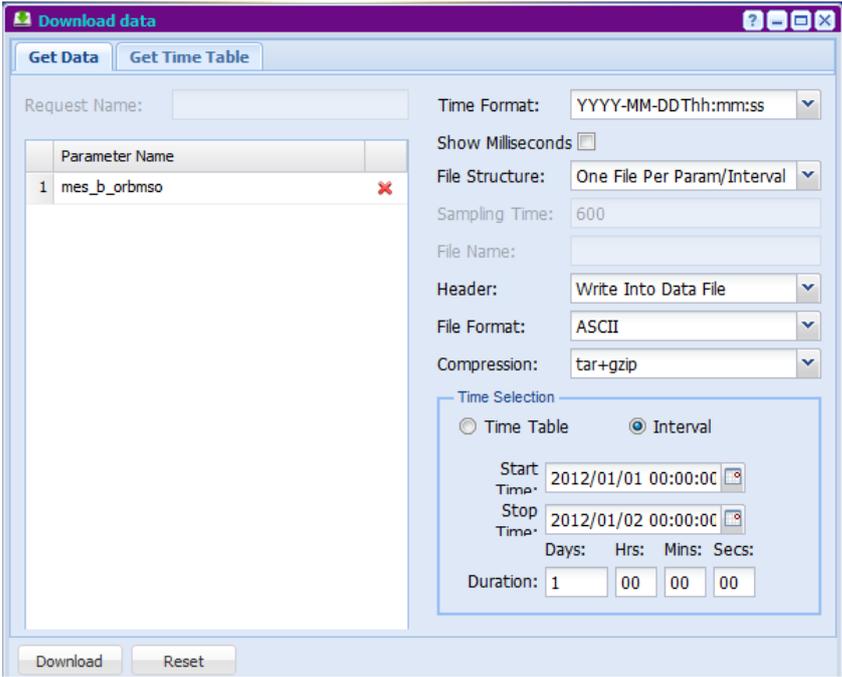
**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Titre du test	Data Mining – Résultat immédiat – Condition « imf(0) > imf(1) »	Numéro de Test	TV-US44-002	
<p>Pré-condition</p>		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Data Mining ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Do Search »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « datamining_XXXX »	OK	Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Edit/Save » pour la TimeTable de l'item « Results ».	L'interface d'édition d'une TimeTable s'ouvre avec les données de la TimeTable calculée	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

4	Cliquez sur le bouton « Edit/Save » pour la Gaps TimeTable de l'item « Results ».	L'interface d'édition d'une Gaps TimeTable s'ouvre avec les données de la TimeTable calculée	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
---	---	--	----	---

5.6.3 Cas test module AMDA_Integration: Batch Mode – US45

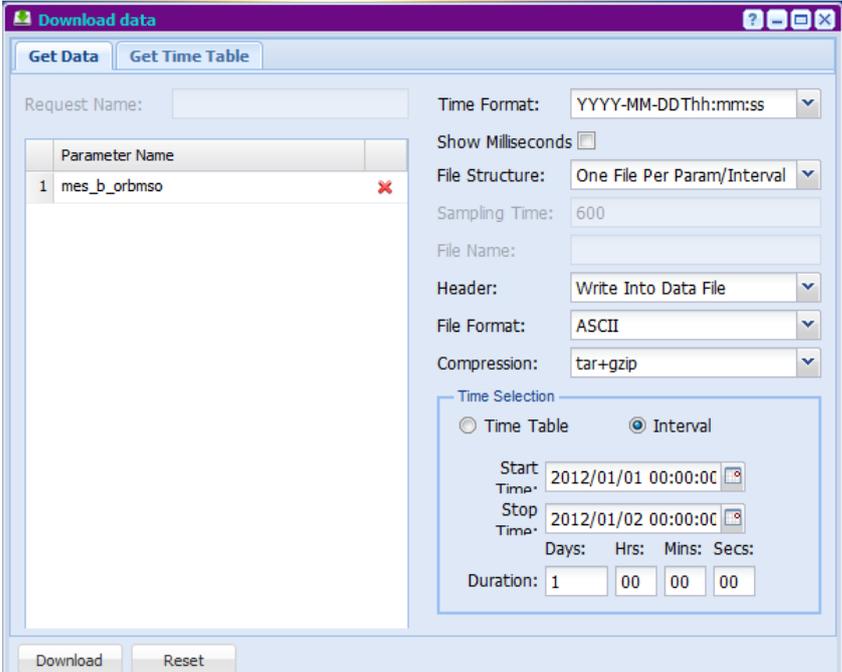
Titre du test	Download - Batch Mode	Numéro de Test	TV-US45-001	
Pré-condition	<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 			
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

2	Attendez le passage du process en « batch mode »	Une fenêtre d'information s'ouvre avec le message « Your request is still running and has been assigned the name download_data_XXX -- Check Jobs in Progress »	OK	Si le résultat est immédiat et ne passe pas par un job, veuillez augmenter l'intervalle de temps de la requête à l'étape 1.
3	Attendez la fin de l'exécution du job	Le job passe du statut « Jobs in Progress » au statut « Finished Jobs » dans l'arbre des jobs	OK	
4	Double cliquez sur le nœud du job terminé dans l'arbre des jobs	La fenêtre « Results » s'affiche avec un item « download_data_XXXX »	OK	
5	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.	OK	Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.
6	Observez l'archive produite	L'archive doit porter l'extension « tar.gz ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir un fichier : « output-mes_b_orbmso_2012000000000000.txt ».	OK	
7	Editez le fichier « output-mes_b_orbmso_2012000000000000.txt »	Le fichier s'ouvre et contient les données demandées.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

Titre du test	Download - Batch Mode – Cancel process	Numéro de Test	TV-US45-002
----------------------	---	-----------------------	--------------------

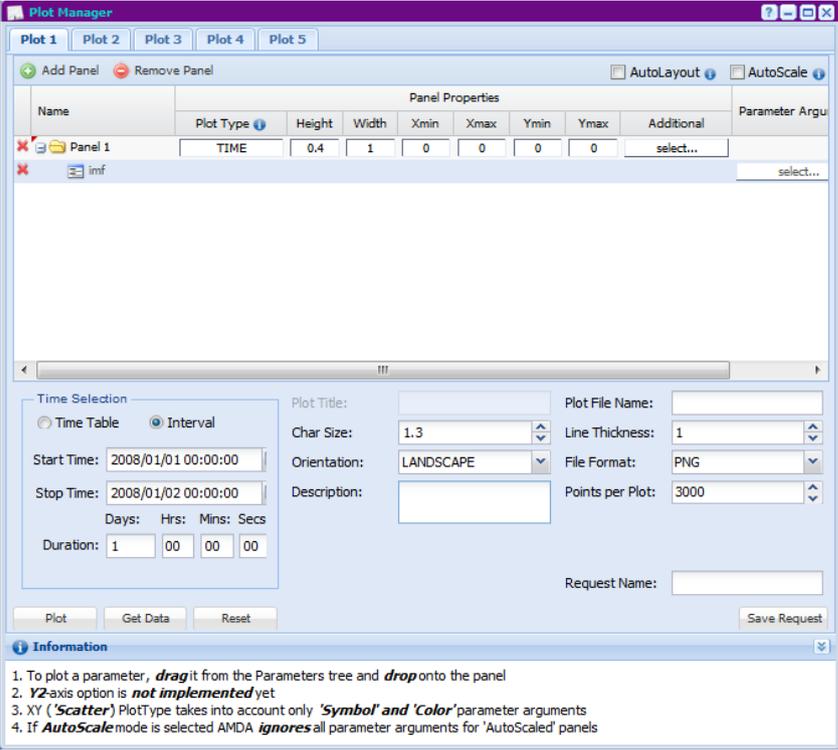
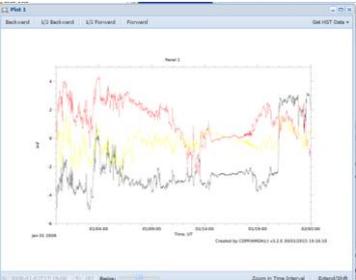
Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Attendez le passage du process en « batch mode »	Une fenêtre d'information s'ouvre avec le message « Your request is still running and has been assigned the name download_data_XXX -- Check Jobs in Progress »	OK	Si le résultat est immédiat et ne passe pas par un job, veuillez augmenter l'intervalle de temps de la requête à l'étape 1.
3	Faites un clic droit sur le nœud du job avant la fin de son exécution (dans le nœud parent 'Jobs In Progress'). Cliquez sur « Stp/Delete Job »	Le Job est supprimé	OK	

5.6.4 Cas test module AMDA_Integration: Plot – US43

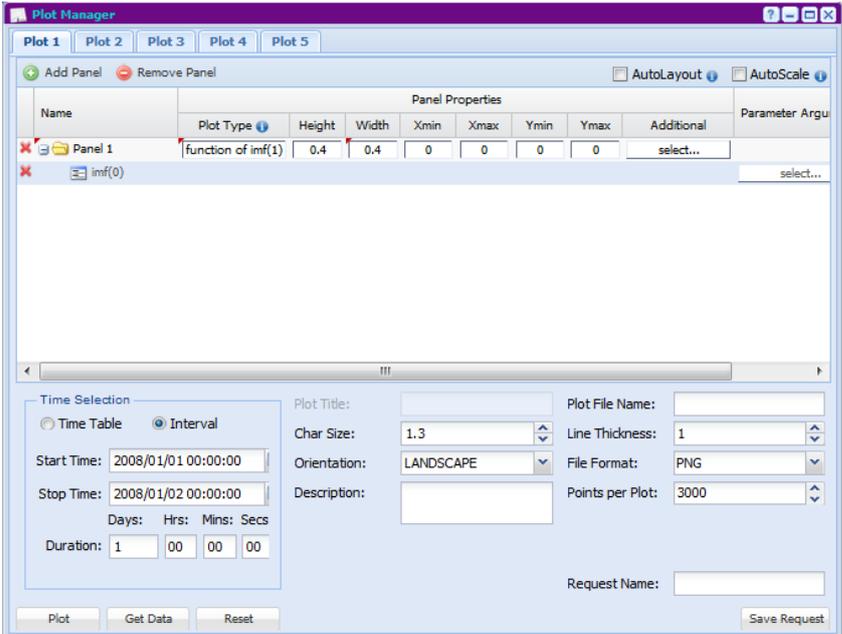
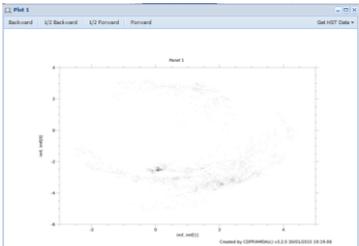
Titre du test	Plot – timePlot	Numéro de Test	TV-US43-001
----------------------	------------------------	-----------------------	--------------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

<p>Pré-condition</p>		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Vérifiez le plot obtenu		OK	

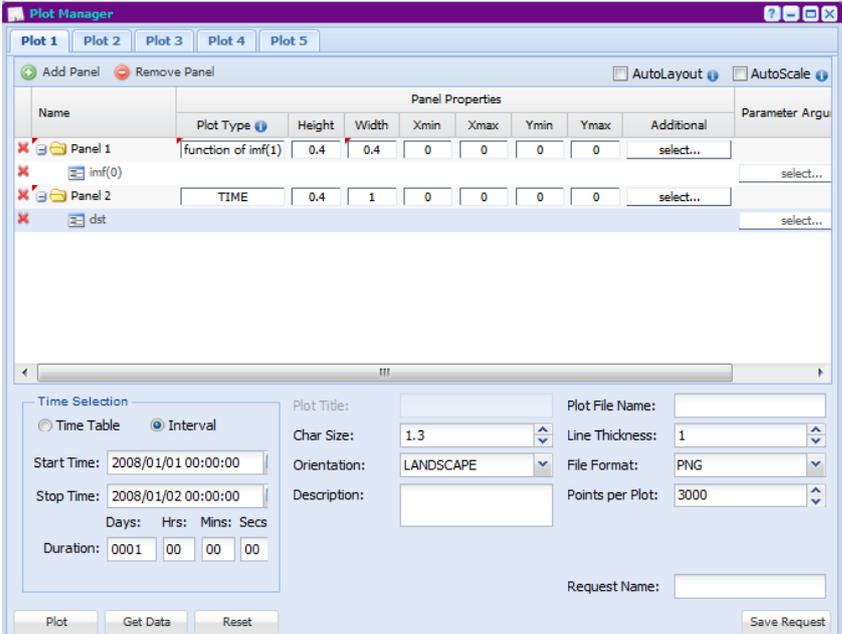
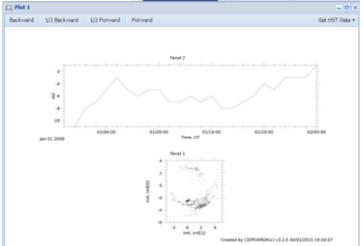
Titre du test	Plot – xyPlot	Numéro de Test	TV-US43-002
---------------	---------------	----------------	-------------

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Pré-condition		Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda). Module « Plot Manager » ouvert et pré-rempli de la manière suivante :		
				
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Vérifiez le plot obtenu		OK	

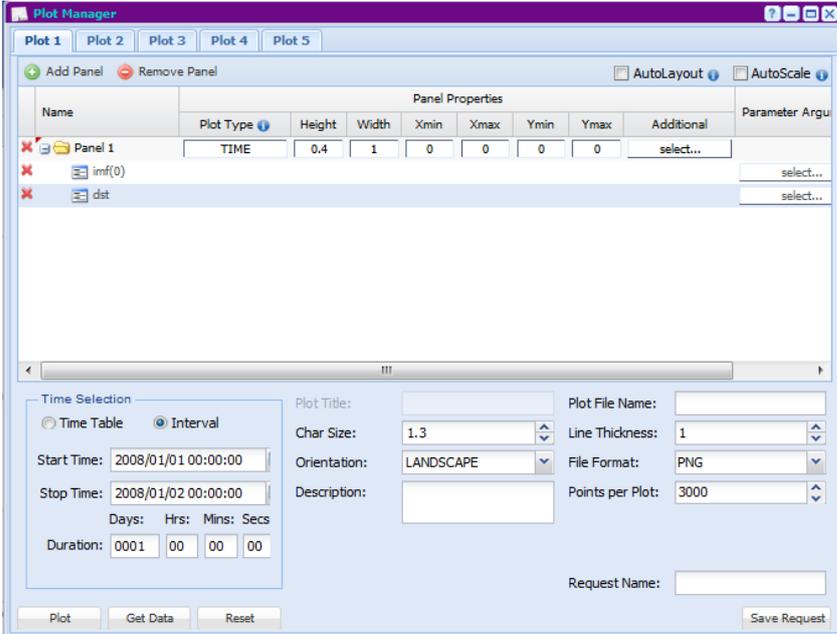
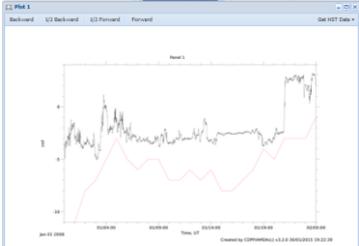
Titre du test	Plot – Deux panels sur la page	Numéro de Test	TV-US43-003
----------------------	---------------------------------------	-----------------------	--------------------

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Vérifiez le plot obtenu		OK	

Titre du test	Plot – Deux séries dans un même panel	Numéro de Test	TV-US43-004
----------------------	--	-----------------------	--------------------

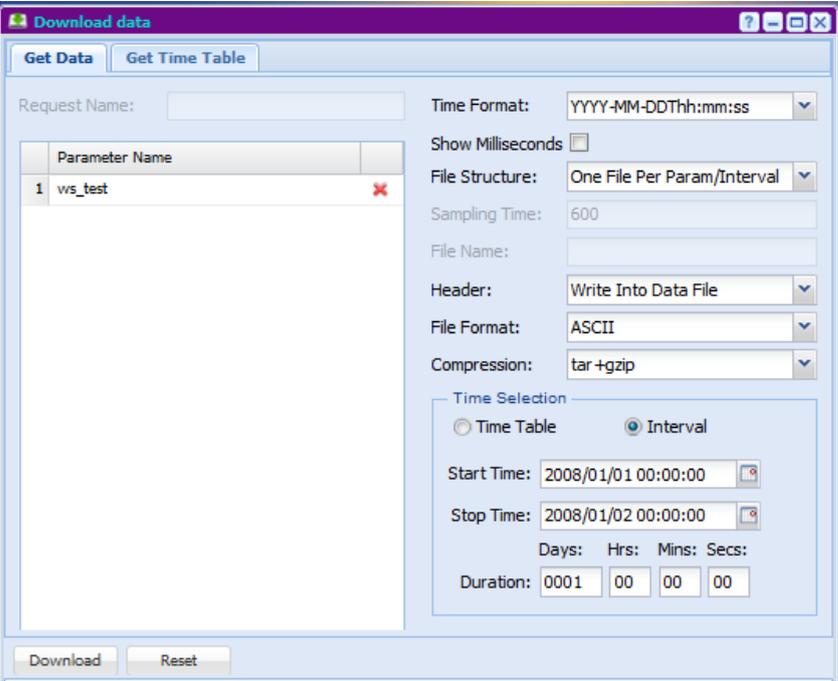
Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Vérifiez le plot obtenu		OK	

5.6.5 Cas test module AMDA_Integration : Paramètre dérivé – US78

Titre du test	Download d'un paramètre dérivé	Numéro de Test	TV-US78-001
----------------------	--------------------------------	-----------------------	-------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

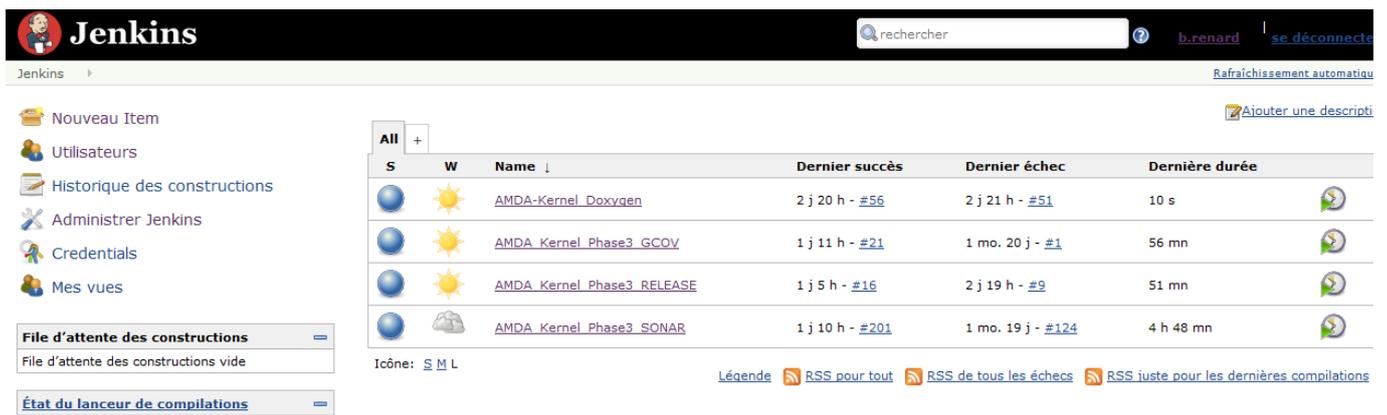
Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Paramètre dérivé suivant créé :</p> <ul style="list-style-type: none"> • Name : test • Time Step : 60 • Construct Parameter : dst+imf(1) <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.	OK	
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »	OK	Si le résultat n'est pas immédiat et passe par un job, veuillez réduire les intervalles de temps de la time table données en requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.	OK	Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.

4	Observez l'archive produite	L'archive doit porter l'extension « tar.gz ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir le fichier « output-ws_test_2008000000000000.txt »	OK	
5	Editez le fichier « output-ws_test_2008000000000000.txt »	Le fichier s'ouvre et contient le header et les données du paramètre dérivé.	OK	Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

5.7 RESULTATS DE TESTS

Pour le module « AMDA_Kernel »: Résultats d'exécution des jobs Jenkins:

- AMDA_Kernel_Phase3_GCOV,
- AMDA_Kernel_Phase3_SONAR,
- AMDA_Kernel_Phase3_RELEASE.



The screenshot shows the Jenkins dashboard with a table of job execution results. The table has columns for status (S), weather icon (W), Name, Dernier succès, Dernier échec, and Dernière durée. The jobs listed are AMDA-Kernel_Doxygen, AMDA_Kernel_Phase3_GCOV, AMDA_Kernel_Phase3_RELEASE, and AMDA_Kernel_Phase3_SONAR.

S	W	Name	Dernier succès	Dernier échec	Dernière durée
		AMDA-Kernel_Doxygen	2 j 20 h - #56	2 j 21 h - #51	10 s
		AMDA_Kernel_Phase3_GCOV	1 j 11 h - #21	1 mo. 20 j - #1	56 mn
		AMDA_Kernel_Phase3_RELEASE	1 j 5 h - #16	2 j 19 h - #9	51 mn
		AMDA_Kernel_Phase3_SONAR	1 j 10 h - #201	1 mo. 19 j - #124	4 h 48 mn

Figure 7 - Résultats d'exécution des jobs Jenkins

Les résultats de tests de validation pour le module « AMDA_Integration » sont présentés dans le tableau des scénarios cf. §5.6.

6 TESTS DE NON-REGRESSION

6.1 METHODOLOGIE DE TESTS DE NON-REGRESSION

Pour le module « **AMDA_Kernel** », tous les tests d'acceptation des Users Stories non obsolètes sont systématiquement repassés via l'outil Jenkins.

Pour le module « **AMDA_Integration** », tous les cas tests sont exécutés manuellement.

Si un des tests ne passe plus après la modification du produit, la priorité est donnée à la résolution de cette régression.

6.2 SCENARIOS DE TESTS

Pour le module « **AMDA_Kernel** » : les scénarios sont décrits par les jobs Jenkins:

- AMDA_Kernel_Phase3_GCOV,
- AMDA_Kernel_Phase3_SONAR,
- AMDA_Kernel_Phase3_RELEASE.

Pour le module « **AMDA_Intégration** » : les scénarios sont décrits sous forme de cas test cf. §5.6.

6.3 RESULTATS DE TESTS

Cf. §5.7

7 MATRICE DE COUVERTURE DES TESTS

Disponible sous IceScrum

8 CAHIER DE RECETTE

Cette section constitue le cahier de recette du projet. Il décrit les modalités de déroulement de la phase de recette en précisant notamment les prérequis, le périmètre, les moyens mis en œuvre, les conditions de passage des tests, les critères d'acceptation et le planning.

8.1 DOMAINE D'APPLICATION

Cette section concerne la recette site.

Les descriptions des tests effectués ainsi que les résultats de ces tests sont décrits :

- Pour le module « AMDA_Kernel » : dans l'outil web de test d'acceptation FitNesse qui sera exécuté durant la recette par l'IRAP/CNES. Une impression pdf de la page de résultat fera office de journal d'essai,
- Pour le module « AMDA_Integration » : les cas tests, ainsi que les résultats attendus, sont présentés dans la section §5.6 et seront exécutés durant la recette par l'IRAP/CNES. Une impression des fiches de tests (cf. §9.2) remplies fera office de journal d'essai.

Le périmètre applicatif concerné est le suivant :

- ☞ Le module « AMDA_Kernel » du logiciel AMDA-NG,
- ☞ Le module « AMDA_Intégration » du logiciel AMDA-NG.

Note : Le module « AMDA_IHM » du logiciel AMDA-NG est à considérer comme un COTS fourni par l'IRAP/CNES en début de prestation. Ce module ne rentre donc pas dans le périmètre applicatif.

8.2 RAPPEL DES FONCTIONNALITES DU PRODUIT

Les fonctionnalités développées en cours de prestation sont présentées dans l'outil Icescrum : <http://bas-agiletools-01.akka.eu:8080/icescrum/p/AKKA3#project>.

8.3 ARCHITECTURE DU PRODUIT

L'architecture du produit telle que fournie par le client en début de prestation est présentée dans [R2], [R3] et [R4].
Le conception mise en œuvre lors de cette prestation s'appuie sur l'ensemble de ces documents de référence et est présentée dans [A1].

8.4 COUVERTURE DE LA RECETTE

La recette couvre :

- L'installation des logiciels. Sa mise en œuvre est décrite dans [A2],
- Les tests d'acceptation du module AMDA_Kernel, et les cas tests du module AMDA_Integration, présentés dans ce document (cf. §5.6). Ces tests permettent de valider le logiciel par rapport aux « Users Stories » (équivalent des exigences fonctionnelles).

La stratégie de validation est entièrement décrite dans ce document, et la stratégie de test pour la recette est de générer, installer puis de rejouer tous les tests (fonctionnels) automatisés (plateforme Fitness) et manuels (cas tests) sur la plateforme cible.

Tous les tests qui seront passés ont été validés lors des diverses revue de Sprint.

8.5 FOURNITURES A RECETTER

8.5.1 Les logiciels

Voir le manuel d'installation [A2] qui présente la liste des logiciels à installer.

Les logiciels à « recetter » se limitent à :

- Module AMDA_Kernel : livré sous forme d'une archive compressée « tgz » avec la nomenclature : AMDA_Kernel_V<version>.tgz,
- Module AMDA_Intégration : : livré sous forme d'une archive compressée « tgz » avec la nomenclature : AMDA_Integration_V<version>.tgz.

Ces archives contiennent l'ensemble des sources de l'application, les tests automatisés ainsi que le client et le serveur de test.

Les données de test sont de la responsabilité de l'IRAP/CNES.

Les COTS sont livrés sous forme d'une archive compressée « tgz » avec la nomenclature : AMDA_COTS_V<version>.tgz.

8.5.2 Les documents

Référence	Nom du document	Version	Date de la version	Emetteur
CDPP-CD-32500-502-SIL	Dossier de conception du noyau AMDA-NG (3 ^{ème} partie) et de son intégration avec l'IHM d'AMDA	V01.2		AKKA IS
CDPP-IF-32500-504-SIL	Dossier de contrôle des interfaces du noyau AMDA-NG (3 ^{ème} partie)	V01.1		AKKA IS
CDPP-MI-32500-505-SIL	Manuel d'installation du noyau AMDA-NG (3 ^{ème} partie) et d'intégration avec l'IHM	V01.2		AKKA IS
CDPP-PE-32500-506-SIL	Noyau AMDA-NG (3 ^{ème} partie) et intégration avec l'IHM AMDA – Plan de test	V01.1		AKKA IS

8.5.3 Les matériels

Sans objet

8.6 CONTEXTE DE LA RECETTE

La recette portera sur toutes les « Users stories » (fonctionnalités) constituant la version du produit livré. Ces « Users stories » ont été acceptées lors des revues de « Sprint ».

Elles sont listées accompagnées de leurs tests d'acceptation dans l'outil web icescrum (<http://bas-agiletools-01.akka.eu:8080/icescrum/p/AKKA3#project>) et seront récapitulées dans le journal des essais.

8.7 LES ESSAIS DE RECETTE

8.7.1 Installation – Paramétrage

Cf. [A2].

Les tests devront être passés sur le produit compilé en mode « Debug » puis en mode « Release ».

8.7.2 Les tests fonctionnels

Cf. §5.6.

Pour le module « AMDA_Kernel », il s'agit des tests automatisés Fitnessse.

Pour le module « AMDA_Intégration », il s'agit d'un ensemble de cas tests.

8.7.3 Les contraintes

8.7.3.1 Contraintes d'environnement

Une instance de DDServer doit être démarrée avec les jeux de données correspondants aux tests d'acceptation des Users Stories disponibles (responsabilité de l'IRAP).

Le server doit pouvoir communiquer/joindre DDServer.

Les pré-requis de [A2] doivent être respectés.

8.7.3.2 Performances

Sans objet

8.8 LOGIQUE D'ENCHAÎNEMENT DES ESSAIS

8.8.1 Dossier de référence

Les tests d'acceptations et les cas tests sont menés sur les versions figées du logiciel :

- Pour le module AMDA_Kernel : la version figée du logicielle est : 3.3.0,
- Pour le module AMDA_Integration : la version figée du logicielle est : 1.2.0.

Ces versions figées dépendent de l'archive des COTS (AMDA_COTS) portant la version 1.1.0.

8.8.2 Liste des tests

Des impressions pdf des résultats FitNesse obtenus en mode Debug et en mode Release, ainsi qu'une impression des fiches de tests (cf. §9.2) remplies constitueront le journal des essais et identifieront clairement la liste des tests passés en recette.

L'IRAP/CNES pourra, dans la limite du planning défini, effectuer des tests supplémentaires. Dans ce cas, la fiche de test devra être rédigée et complétée par l'IRAP/CNES sous IceScrum et FitNesse, puis consignée dans le journal des essais.

8.9 ORGANISATION DE LA RECETTE

Les machines utilisées pour les tests de recette sont les suivantes :

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Code	Type de machine	Système d'exploitation	Description
cdpp1.cesr.fr	Dell PowerEdge R620 Bi-proc intel Xeon E5- 2640 2.50GHz – 15M Cache – Turbo 6C – 48Go de mémoire	Cent OS 6.3 64bits	Machine située à l'IRAP. Le noyau n'est pas hébergé sur la machine cdpp1.cesr.fr, mais sur une machine virtuelle Virtualbox (qui elle tourne sur cdpp1), dont voici les caractéristiques : <ul style="list-style-type: none"> • amdakernel.cesr.fr • Système d'exploitation Cent OS 6.3 64 bits • 1 processeur utilisé • 1 Go de mémoire allouée Cette machine virtuelle contient : <ul style="list-style-type: none"> • Le noyau • Les outils de développement pour le noyau La plateforme de test fitnessse
manunja.cesr.fr			Machine hébergeant DDServer

Les outils logiciels suivants sont nécessaires :

Outil	Version	Plate-forme
Firefox	Sans grande importance Version 33.1 testée	Machine accédant au FitNesse de la machine d'intégration

Moyens d'essais : Sans Objet.

8.10 PLANNING

La recette de la version 3.3.0 du module AMDA_Kernel, et de la version 1.2.0 du module AMDA_Integration se déroule sur **5 jours ouvrés**.

L'ensemble des tests prévus est déroulé sur cette période. Ce délai ne peut être remis en cause si l'ensemble des tests prévus n'a pu être passé complètement. Toutefois, des anomalies bloquantes pour les tests peuvent ajourner la recette et nécessiter de poursuivre celle-ci une fois les corrections apportées.

8.11 INTERVENANTS

L'IRAP procédera à la recette.

8.12 RESULTATS DE LA RECETTE

8.12.1 Conditions d'acceptation

Suite à la livraison complète des éléments de recette, l'IRAP/CNES dispose d'un délai fixe pour effectuer la recette. Ce délai est défini pour chaque version recettée en fonction du périmètre fonctionnel soumis à acceptation.

En fin de période de recette, l'analyse des anomalies résiduelles permet de donner les éléments de prononciation ou non de la recette :

- S'il subsiste des Anomalies Bloquantes, la prononciation de la recette peut être refusée ;
- S'il subsiste des Anomalies non-bloquantes, la recette est prononcée avec réserves et conditionnée par un plan d'actions de corrections, annexé au « Procès Verbal de Recette » ;
- Sinon, la recette est prononcée sans réserve

Une fois les anomalies corrigées et après recette par l'IRAP/CNES, le responsable de la réalisation procède à la livraison définitive (bon de livraison) dans un espace dédié pour mise en exploitation.

Les Anomalies Externes ne peuvent pas entraîner de refus de recette.

Anomalie externe : anomalie due à une cause étrangère à la fourniture de AKKA. Dans cette définition d'anomalie entrent par exemple :

- les erreurs consécutives aux éléments manquants autres que ceux dont AKKA a la charge,
- les anomalies relatives aux pannes matérielles,
- les anomalies liées à une mauvaise configuration (variables d'environnement, fichiers de paramétrage).

NB : Une mise en exploitation entraîne la prononciation de la recette.

8.12.2 Gestion des anomalies détectées en recette

En cas d'anomalie détectée lors de la recette, une « user story » de type « défaut » est créée sous IceScrum avec le test qui a remonté cette anomalie. Un plan d'action est alors proposé par l'équipe AKKA Technologies.

En cas d'anomalie bloquante, la recette peut être ajournée, le temps pour AKKA Technologies de corriger l'anomalie et livrer un patch correctif.

9 ANNEXE

9.1 ANALYSE DU RAPPORT SONAR

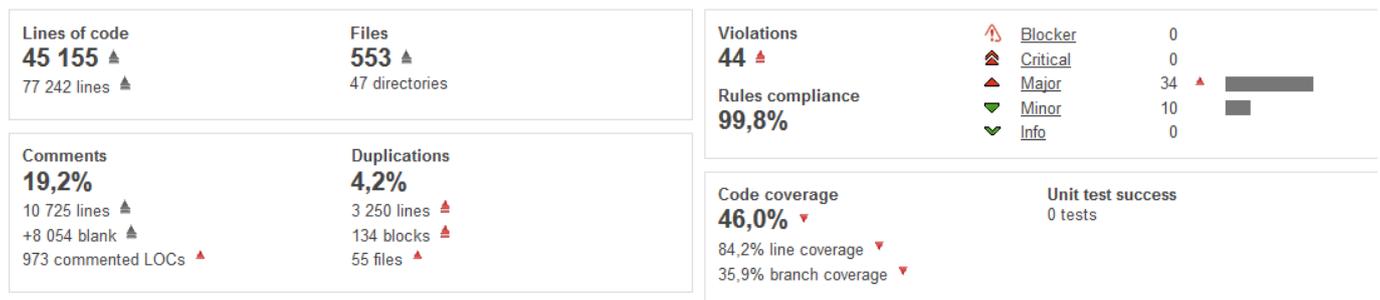


Figure 8 - Rappel du rapport Sonar

9.1.1 Commentaires

Le taux de commentaire est de 19.2%.

9.1.2 Duplications

Le taux de duplication de code est de 4.2%.

Près de la moitié des duplications proviennent des fichiers « .hh » des implémentations du FileWriter qui héritent de la classe « FileWriterASCIIAbstract » au niveau du module « Download ».

Par exemple, le code présenté ci-dessous est présent dans les fichiers « FileWriterASCIIVOTable.hh », « FileWriterASCIIJson.hh » et « FileWriterASCIITabular.hh » :

```

/*
 * @overload FileWriterASCIIAbstract::getExtension - Method to get file format extension
 */
virtual std::string getExtension(void);

/*
 * @overload FileWriterASCIIAbstract::getDataFillCharacter Method to get fill character to use

```

Cela n'est pas choquant de la part de part de classes qui héritent d'une classe abstraite commune.

Une autre partie des duplications proviennent de l'héritage des prestations précédentes.

Enfin, les autres duplications détectées nous semblent acceptables pour un projet de cette complexité.

9.1.3 Violations majeures

34 violations majeures sont détectées. Elles correspondent en fait à une seule violation répétée lors de l'exécution de plusieurs tests (tous les tests de plot contenant un color bar).

La violation en question est une violation de la librairie PLPlot lors de l'appel à la fonction « colorbar ». Il ne s'agit donc pas d'une violation provoquée par le module « AMDA_Kernel ».

9.1.4 Violations mineures

Dix violations mineures sont détectées :

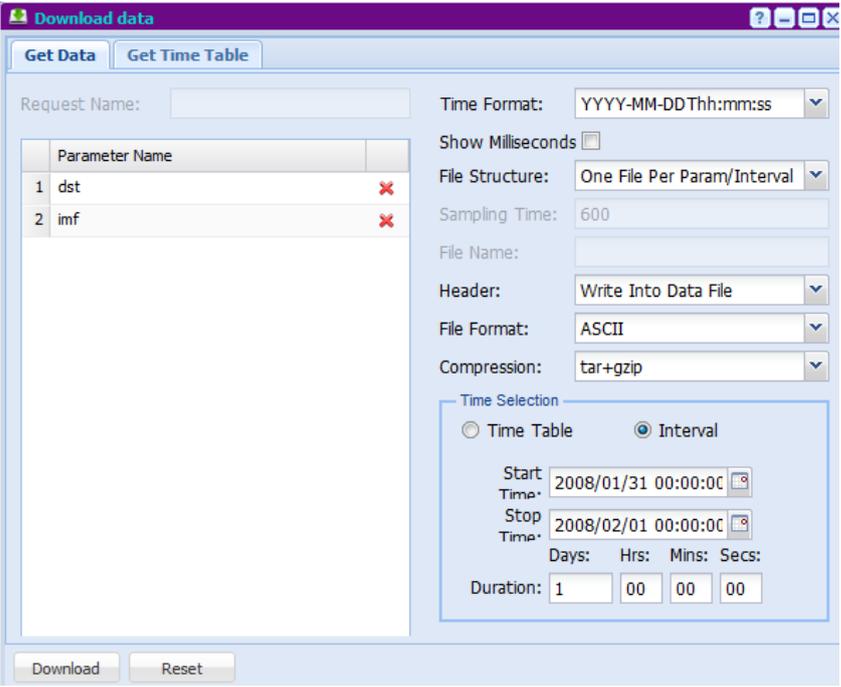
- Sept sont du type « Function never used ». Il s'agit de faux positifs pour des méthodes appelées par des Macros,
- Trois sont du type « Class does not have a constructor although it has private member variables » dans le fichier « Resampling.hh » et dans le fichier « MinMaxMeanStatistic.hh ». Après vérification, nous estimons ces violations comme étant de faux positifs.

9.1.5 Couverture du code par les tests

35.9% des branches du code sont couvertes selon GCOV, alors que 84.2% des lignes de code le sont.

9.2 FICHES DE TESTS

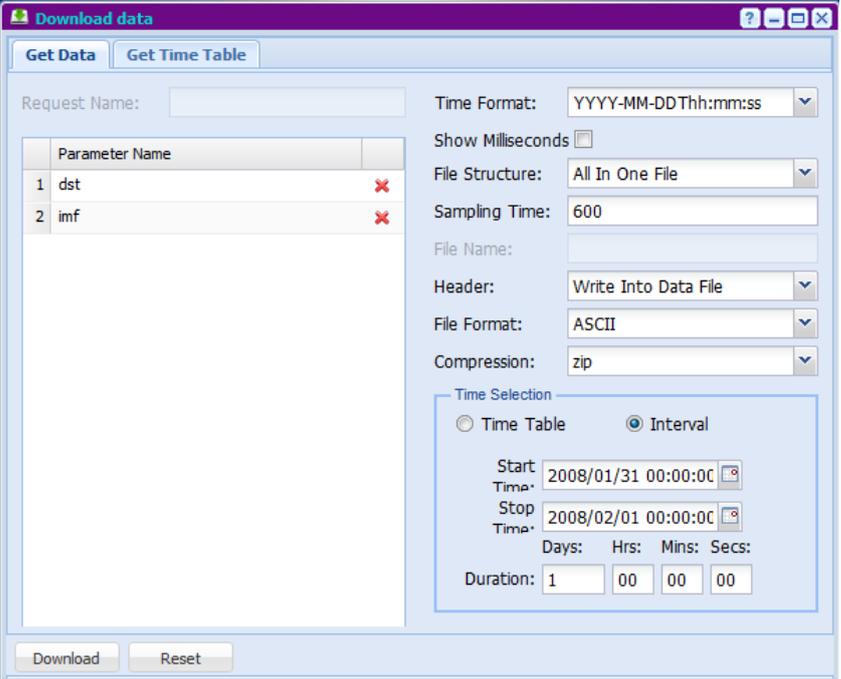
9.2.1 Cas test module AMDA_Integration: Download – Résultat immédiat – US42

Titre du test	Download – Résultat immédiat – Deux paramètres – Structure « One File Per Param/Interval » - Compression « tar+gzip »	Numéro de Test	TV-US42-001
Pré-condition	<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 		

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »		Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.		Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.
4	Observez l'archive produite	L'archive doit porter l'extension « tar.gz ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir deux fichiers : « output-dst_2008030000000000.txt » « output-imf_2008030000000000.txt »		
5	Editez le fichier « output-dst_2008030000000000.txt »	Le fichier s'ouvre et contient un header et des données. Le temps d'échantillonnage est 3600s.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
6	Editez le fichier « output-imf_2008030000000000.txt »	Le fichier s'ouvre et contient un header et des données. Le temps d'échantillonnage est 16s.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

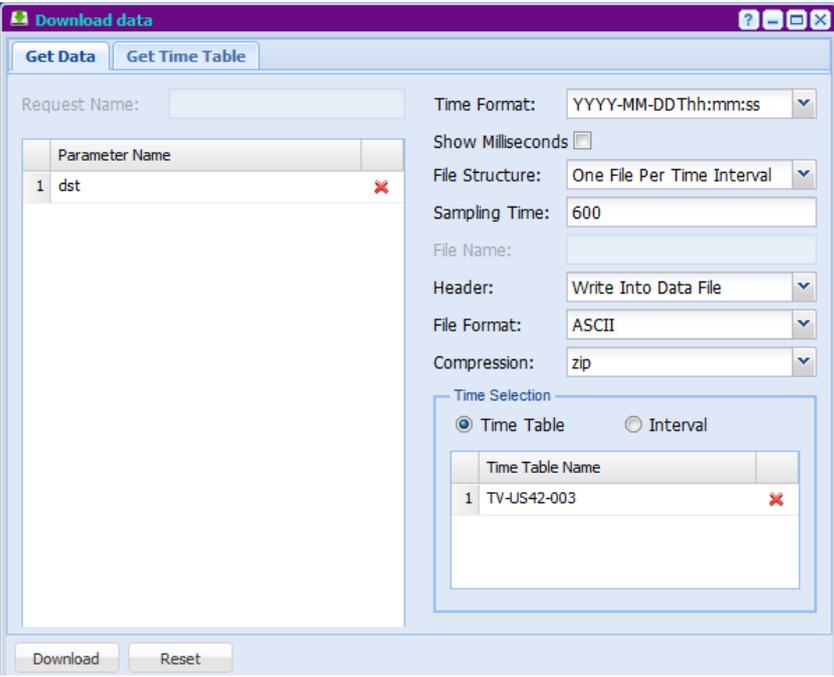
Titre du test	Download – Résultat immédiat – Deux paramètres – Structure « All In One File » - Sampling time « 600 » - Compression « zip »	Numéro de Test	TV-US42-002	
Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »		Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.		Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

4	Observez l'archive produite	<p>L'archive doit porter l'extension « zip ».</p> <p>Elle doit être « ouvrable » par votre outil de décompression.</p> <p>Elle doit contenir un seul fichier :</p> <p>« output-dst_imf_2008030000000000.t xt »</p>		
5	Editez le fichier « output-dst_imf_2008030000000000.t xt »	<p>Le fichier s'ouvre et contient un header et des données.</p> <p>Le temps d'échantillonnage est 600s.</p>		<p>Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.</p>

Titre du test	Download – Résultat immédiat – Un paramètre – Structure « One File Per Time Interval » - Sampling time « 600 » - Compression « zip »	Numéro de Test	TV-US42-003
----------------------	---	-----------------------	--------------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

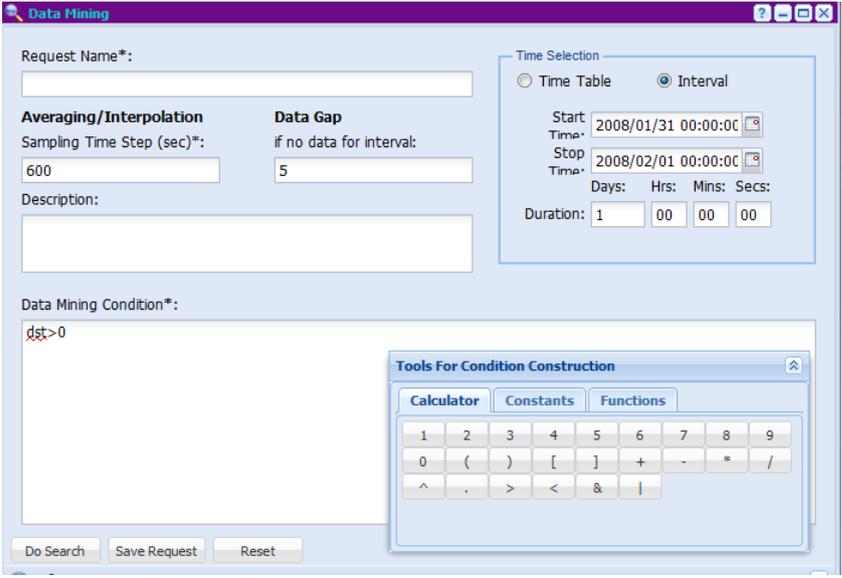
Pré-condition	<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>TimeTable créée portant le nom « TV-US42-003 » et contenant les intervalles de temps : « 2008-01-31T00:00:00 - 2008-01-31T04:00:00 » et « 2008-01-31T08:00:00 - 2008-01-31T12:00:00 »</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 			
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »		Si le résultat n'est pas immédiat et passe par un job, veuillez réduire les intervalles de temps de la time table données en requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.		Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

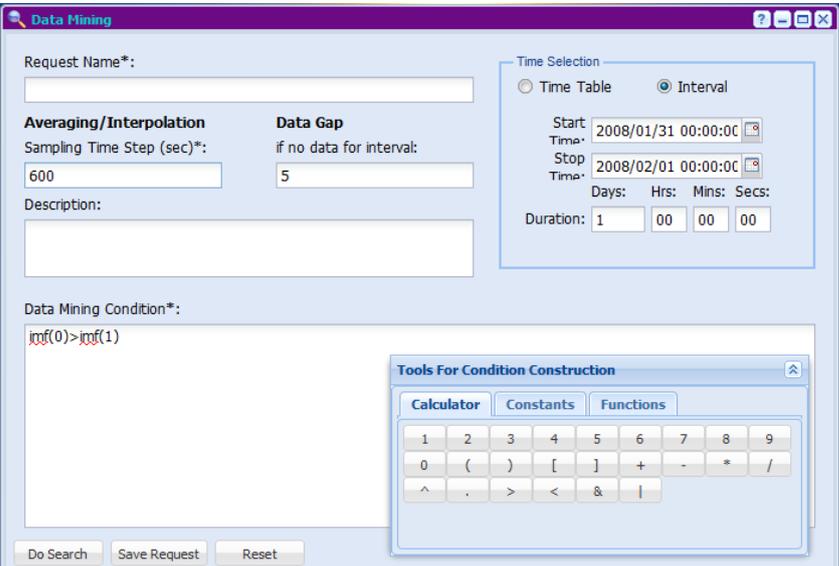
4	Observez l'archive produite	L'archive doit porter l'extension « zip ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir trois fichiers : « output-dst_0.txt », « output-dst_1.txt » et « output-dst_info.txt »		
5	Editez le fichier « output-dst_info.txt »	Le fichier s'ouvre et contient uniquement le header.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
6	Editez le fichier « output-dst_0.txt »	Le fichier s'ouvre et contient les données relatives au premier intervalle de temps.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
7	Editez le fichier « output-dst_1.txt »	Le fichier s'ouvre et contient les données relatives au deuxième intervalle de temps.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

9.2.2 Cas test module AMDA_Integration: Data Mining – Résultat immédiat – US44

Titre du test	Data Mining – Résultat immédiat – Condition « dst > 0 »	Numéro de Test	TV-US44-001
----------------------	---	-----------------------	--------------------

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Data Mining ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Do Search »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez la fin d'exécution de la requête	A l'issu de l'exécution, la fenêtre « Results » s'affiche avec un item « datamining_XXXX »		Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Edit/Save » pour la TimeTable de l'item « Results ».	L'interface d'édition d'une TimeTable s'ouvre avec les données de la TimeTable calculée		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
4	Cliquez sur le bouton « Edit/Save » pour la Gaps TimeTable de l'item « Results ».	L'interface d'édition d'une Gaps TimeTable s'ouvre avec les données de la TimeTable calculée		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

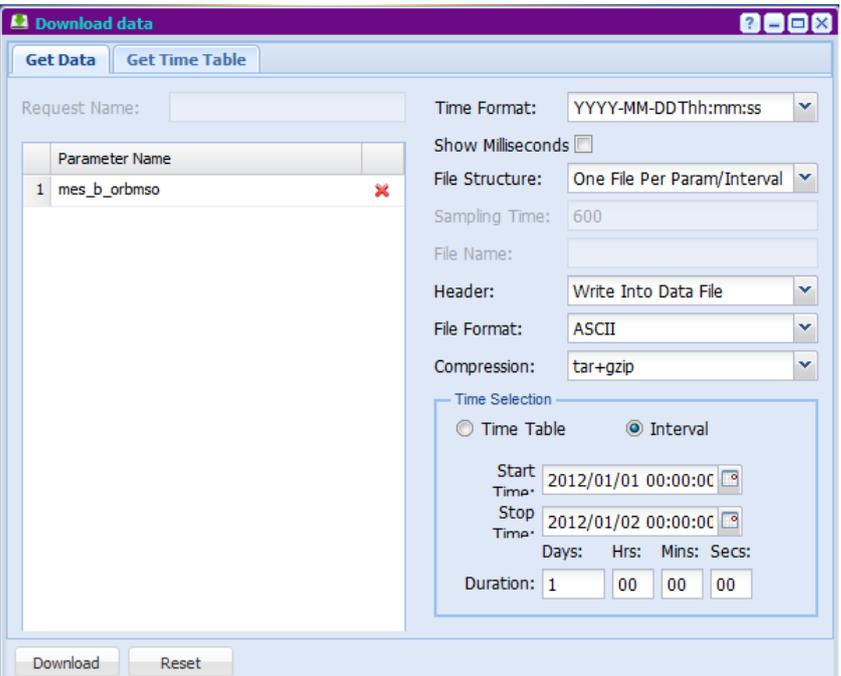
**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Titre du test	Data Mining – Résultat immédiat – Condition « imf(0) > imf(1) »	Numéro de Test	TV-US44-002	
Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Data Mining ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Do Search »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « datamining_XXXX »		Si le résultat n'est pas immédiat et passe par un job, veuillez réduire l'intervalle de temps de la requête à l'étape 1.
3	Cliquez sur le bouton « Edit/Save » pour la TimeTable de l'item « Results ».	L'interface d'édition d'une TimeTable s'ouvre avec les données de la TimeTable calculée		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

4	Cliquez sur le bouton « Edit/Save » pour la Gaps TimeTable de l'item « Results ».	L'interface d'édition d'une Gaps TimeTable s'ouvre avec les données de la TimeTable calculée		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.
---	---	--	--	---

9.2.3 Cas test module AMDA_Integration: Batch Mode – US45

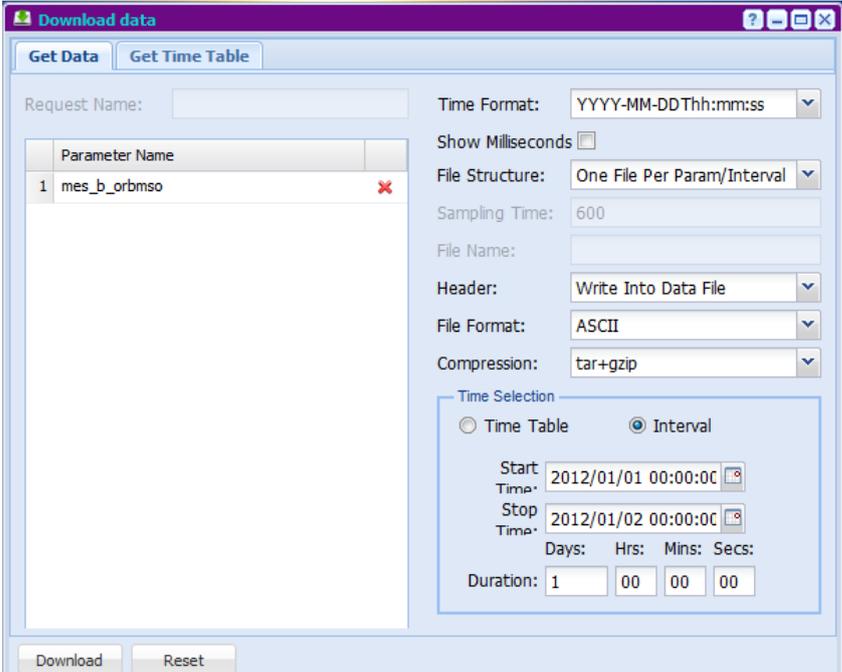
Titre du test	Download - Batch Mode	Numéro de Test	TV-US45-001	
<p>Pré-condition</p> <p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p>				
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

2	Attendez le passage du process en « batch mode »	Une fenêtre d'information s'ouvre avec le message « Your request is still running and has been assigned the name download_data_XXX -- Check Jobs in Progress »		Si le résultat est immédiat et ne passe pas par un job, veuillez augmenter l'intervalle de temps de la requête à l'étape 1.
3	Attendez la fin de l'exécution du job	Le job passe du statut « Jobs in Progress » au statut « Finished Jobs » dans l'arbre des jobs		
4	Double cliquez sur le nœud du job terminé dans l'arbre des jobs	La fenêtre « Results » s'affiche avec un item « download_data_XXXX »		
5	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.		Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.
6	Observez l'archive produite	L'archive doit porter l'extension « tar.gz ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir un fichier : « output-mes_b_orbmso_2012000000000000.txt ».		
7	Editez le fichier « output-mes_b_orbmso_2012000000000000.txt »	Le fichier s'ouvre et contient les données demandées.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

Titre du test	Download - Batch Mode – Cancel process	Numéro de Test	TV-US45-002
----------------------	---	-----------------------	--------------------

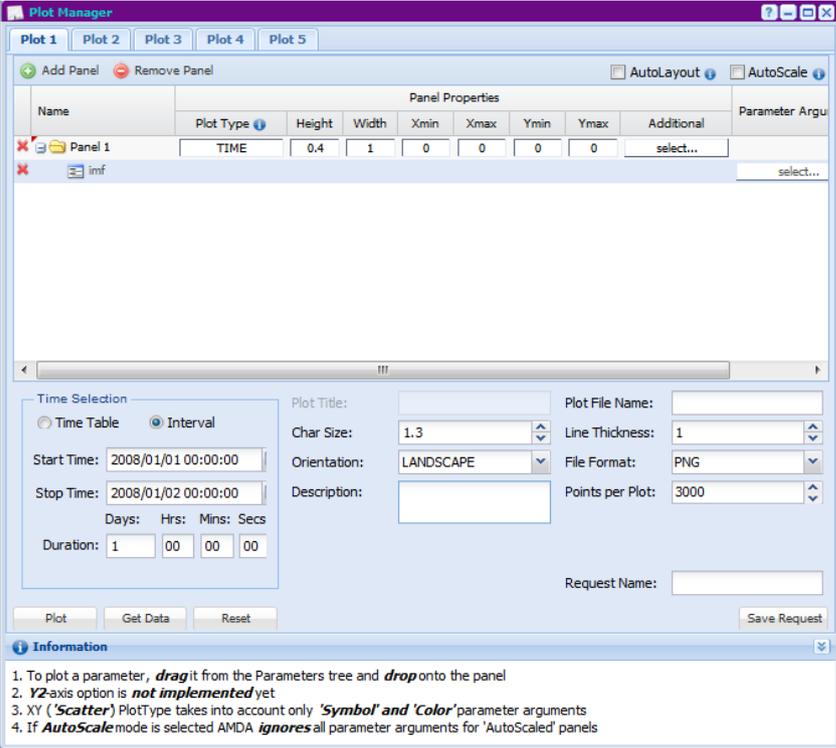
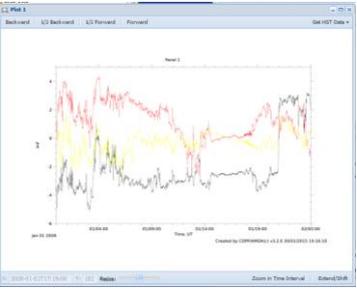
Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module Download ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez le passage du process en « batch mode »	Une fenêtre d'information s'ouvre avec le message « Your request is still running and has been assigned the name download_data_XXX -- Check Jobs in Progress »		Si le résultat est immédiat et ne passe pas par un job, veuillez augmenter l'intervalle de temps de la requête à l'étape 1.
3	Faites un clic droit sur le nœud du job avant la fin de son exécution (dans le nœud parent 'Jobs In Progress'). Cliquez sur « Stp/Delete Job »	Le Job est supprimé		

9.2.4 Cas test module AMDA_Integration: Plot – US43

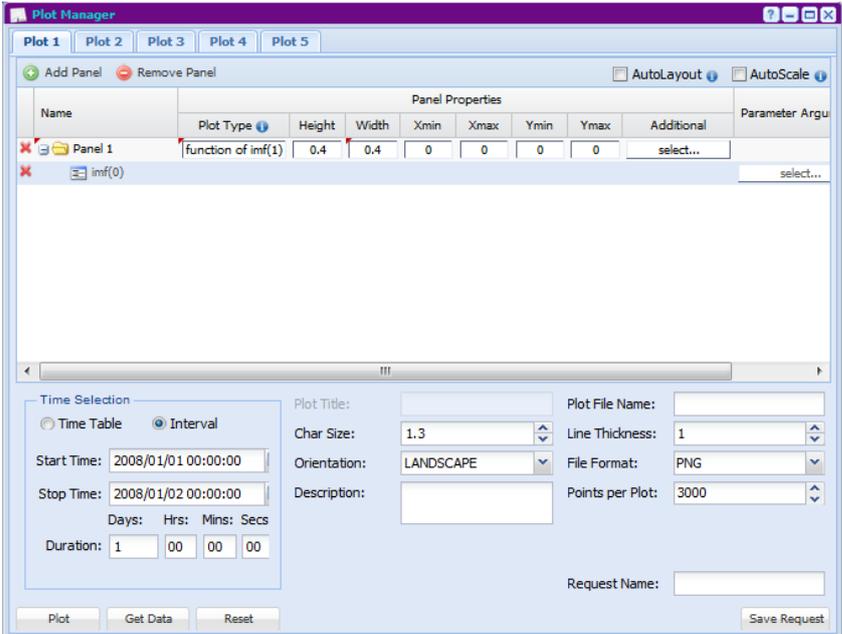
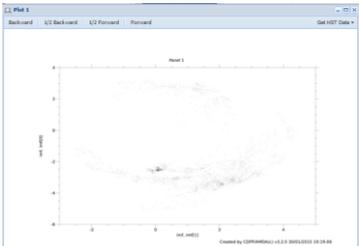
Titre du test	Plot – timePlot	Numéro de Test	TV-US43-001
----------------------	------------------------	-----------------------	--------------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition	<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 			
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Vérifiez le plot obtenu			

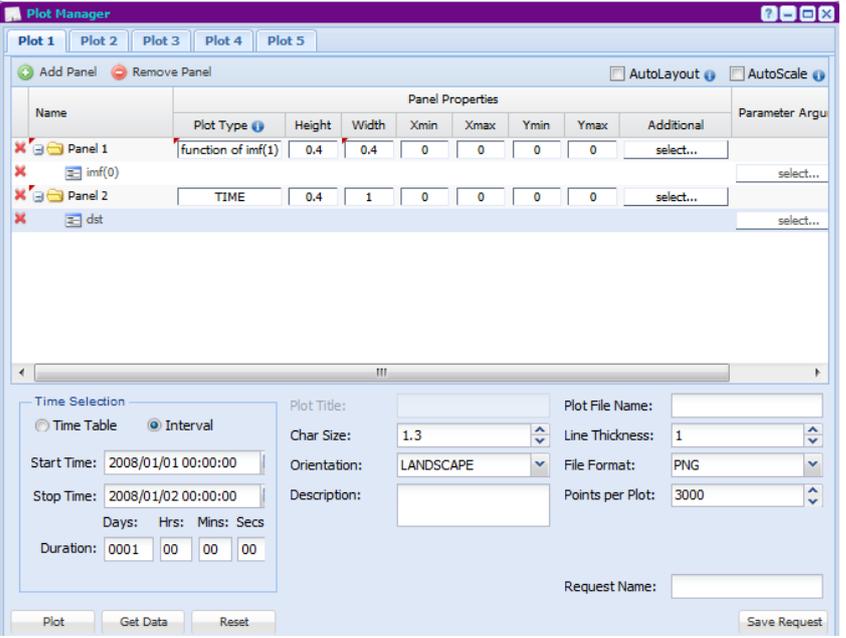
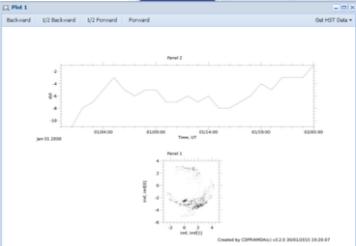
Titre du test	Plot – xyPlot	Numéro de Test	TV-US43-002
----------------------	---------------	-----------------------	-------------

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

Pré-condition		Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda). Module « Plot Manager » ouvert et pré-rempli de la manière suivante :		
				
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Vérifiez le plot obtenu			

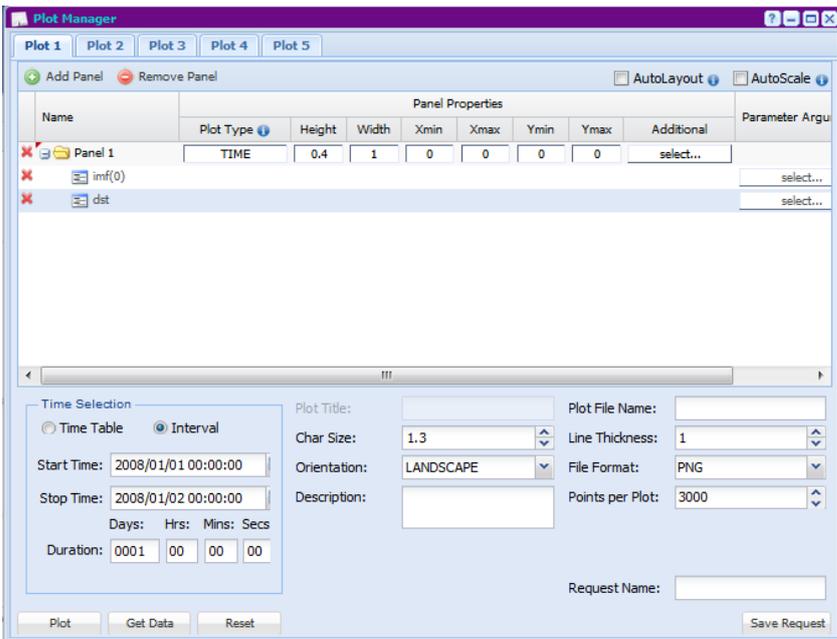
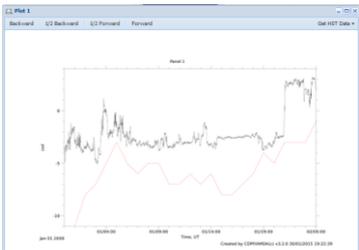
Titre du test	Plot – Deux panels sur la page	Numéro de Test	TV-US43-003
----------------------	---------------------------------------	-----------------------	--------------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

<p>Pré-condition</p>		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Vérifiez le plot obtenu			

Titre du test	Plot – Deux séries dans un même panel	Numéro de Test	TV-US43-004
----------------------	--	-----------------------	--------------------

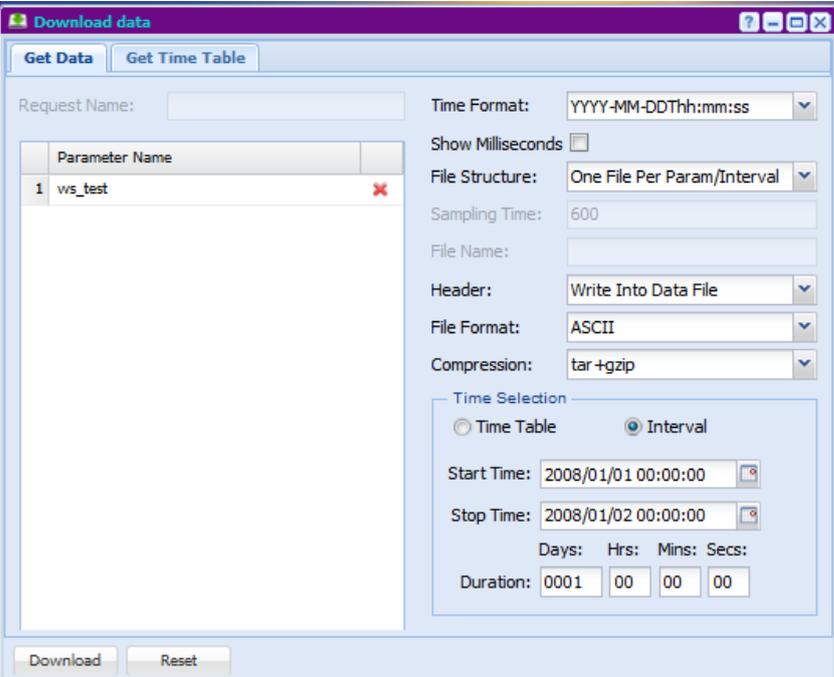
Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Plot »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Vérifiez le plot obtenu			

9.2.5 Cas test module AMDA_Integration : Paramètre dérivé – US78

Titre du test	Download d'un paramètre dérivé	Numéro de Test	TV-US78-001
----------------------	--------------------------------	-----------------------	-------------

Noyau AMDA-NG (3ème partie) et intégration avec l'IHM AMDA - Plan de test

Pré-condition		<p>Utilisateur connecté dans l'interface AMDA (user : testKernel, mdp : amda).</p> <p>Paramètre dérivé suivant créé :</p> <ul style="list-style-type: none"> • Name : test • Time Step : 60 • Construct Parameter : dst+imf(1) <p>Module « Plot Manager » ouvert et pré-rempli de la manière suivante :</p> 		
Numéro d'étape	Description de l'action utilisateur	Résultat attendu	Résultat constaté	commentaire
1	Cliquez sur le bouton « Download »	L'exécution de la requête de lance. Le message d'attente « Processing... Please wait... » s'affiche.		
2	Attendez la fin d'exécution de la requête	A l'issue de l'exécution, la fenêtre « Results » s'affiche avec un item « download_data_XXXX »		Si le résultat n'est pas immédiat et passe par un job, veuillez réduire les intervalles de temps de la time table données en requête à l'étape 1.
3	Cliquez sur le bouton « Download » de l'item « Results ».	Le téléchargement de l'archive produite se lance.		Suivant votre navigateur et sa configuration, une étape de validation du téléchargement peut s'afficher.

**Noyau AMDA-NG (3ème partie) et intégration avec l'IHM
AMDA - Plan de test**

4	Observez l'archive produite	L'archive doit porter l'extension « tar.gz ». Elle doit être « ouvrable » par votre outil de décompression. Elle doit contenir le fichier « output-ws_test_2008000000000000.txt »		
5	Editez le fichier « output-ws_test_2008000000000000.txt »	Le fichier s'ouvre et contient le header et les données du paramètre dérivé.		Les données fournies sont générées par le module « AMDA_Kernel » et sont donc testées par les tests de validation de ce module.

10 DOCUMENTS APPLICABLES ET DE REFERENCE (A/R)

A/R	Référence	Titre
R1	RNC-CNES-Q-HB-80-513	Règles pour l'utilisation du langage C++
R2	CDPP-AR-32500-382-SI	Dossier d'architecture du noyau d'AMDA-NG
R3	CDPP-CD-32500-436-SI	Dossier de conception du noyau d'AMDA-NG
R4	CDPP-CD-32500-457-CS	Dossier de conception – Noyau AMDA-NG – 2 nd e partie
A1	CDPP-CD-32500-502-SIL	Dossier de conception du noyau AMDA-NG (3 ^{ème} partie) et de son intégration avec l'IHM AMDA
A2	CDPP-MI-32500-505-SIL	Manuel d'installation du noyau AMDA-NG (3 ^{ème} partie) et d'intégration avec l'IHM

11 GLOSSAIRE ET ABREVIATIONS

11.1 GLOSSAIRE

Terme	Définition
Campagne de tests	Période continue d'activités de tests, comprenant tout ou partie des scénarios de tests
Cas de test	Un cas de tests est défini par l'état des données en entrée du test, les actions à mener par le testeur et les résultats attendus.
Jeux de tests	<p>Il s'agit de créer des cas concrets d'utilisation de l'application à partir :</p> <ul style="list-style-type: none"> de sa description formalisée dans les spécifications des règles définies dans le plan et dossier de tests (principalement en matière d'ergonomie) <p>La définition de jeux de tests permet de prévoir et de formaliser la manière et les critères sur lesquels l'application va être recettée. Les jeux de tests ne peuvent être définis qu'une fois les spécifications validées.</p>
Jeux de données	Ensemble de données de tests utilisées dans le même scénario (ou jeux d'essai)
Scénario de tests	Enchaînement chronologique de cas de tests, cohérent fonctionnellement et dont le résultat est un état stable du système.
Test boîte blanche (ou test structurel)	Méthode de test qui consiste à concevoir les données d'entrée et les résultats attendus en examinant la structure interne de l'objet à tester.
Test boîte noire (ou test fonctionnel)	Méthode de test qui consiste à concevoir les données d'entrée et les résultats attendus à partir des fonctions spécifiées de l'objet à tester, sans examiner sa structure interne.
Test de bon fonctionnement	<p>Les tests de bon fonctionnement font partie des tests de validation.</p> <p>Il s'agit de tests effectués sur l'environnement cible (ou un environnement de simulation) et qui consistent à s'assurer que le système fonctionne normalement dans cet environnement. Au minimum, les tests de bon fonctionnement consistent à lancer l'application et à s'assurer qu'il s'exécute normalement sans blocage.</p> <p>Les tests de bon fonctionnement peuvent aussi consister à repasser tout ou partie des tests de validation effectués sur l'environnement de tests, voire des tests complémentaires nécessités par des différences entre les environnements.</p>
Test d'intégration	Activité permettant de vérifier que les interfaces communes à plusieurs composants permettent bien de réaliser le comportement attendu entre ces composants. Les tests d'intégration répondent aux exigences de conception.
Test unitaire	Activité permettant de vérifier qu'un composant, pris isolément, satisfait à ses exigences fonctionnelles et techniques. Ils répondent à la conception détaillée si elle a été réalisée.

Terme	Définition
Test de validation	Activité permettant de vérifier que le logiciel, dans son ensemble, satisfait à ses exigences fonctionnelles et techniques y compris dans son environnement cible. Les tests d'intégration répondent aux exigences de spécification.

11.2 ABREVIATIONS

Abréviation	Nom détaillé
DT	Dossier de tests
FT	Fiche de test
MCE	Matrice de couverture des exigences