

\*\*\*\*\* Document global des tests \*\*\*\*\*

XXX
X
X RAPPORT GLOBAL DES TESTS DU LOGICIEL portageMM2CPP X
X X
XXX

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

XXX
X
X Nom du fichier : rapportGlobalDesTests
X
X Présentation du fichier :
X
X Ce fichier de documentation est un recueil d'informations sur les tests du
X logiciel portageMM2CPP.
X
X Ce fichier a été généré automatiquement par exécution de la commande :
X genererLeRapportGlobalDesTests rapportGlobalDesTests.
X Il contient des informations recopiées d'ailleurs.
X
X Pour la présentation des tests du logiciel, voir le fichier
X help\_testsDuLogiciel.
X
X Ici, dans le fichier rapportGlobalDesTests, sont regroupés les
X rapports des tests de chaque scénario de test.
X
X Plus exactement, le fichier contient pour chaque scénario de test scnXXX, les
X uns à la suite des autres :
X - le nom du scénario : \*\*\* SCENARIO DE TEST scnXXX.
X - sa PARTIE DESCRIPTION DE TEST.
X - sa PARTIE RAPPORT DE TEST contenant date après date le rapport des
X tests effectués.
X - le Statut du SCENARIO DE TEST scnXXX.
X
XXX

XXX

\*\*\* SCENARIO DE TEST scnl nominal : fait partie des tests vérifiant que le logiciel p
ortageMM2CPP écrit en C++ (leSourceCPP) est conforme au modèle sous sa forme ModelMa
ker (modele2.mod) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* \* \* \* \* \* Rapport de test \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*  
Description du test :

Les vérifications du scénario scnlnominal ne portent que sur les entités présentes dans le modèle ModelMaker modele2.mod, c'est à dire sur le modèle dans son état jusqu'à la version V071026.

Notamment les tests ne prennent pas en considération les entités qui ont été ajoutées (en plus des entités du modèle ModelMaker modele2.mod) relativement aux modifications m\_linearInterpolation et m\_integrale (entités controlled, flux, compartiments). Pour les tests relatifs aux modifications m\_linearInterpolation et m\_integrale, voir respectivement scnInterpolation et scnIntegrale.

Informations sur la simulation effectuée dans ModelMaker :

- La simulation effectuée : voir lesEntreesTest/modele2.mod.
- Les données résultats : voir "lesEntreesTest".

Informations sur la simulation effectuée en C++ :

- La simulation effectuée : voir mainTestPortageMM2CPP.cpp.
- Les données résultats : voir "lesSortiesTest".

\*\*\*\*\*  
Jeu de test JEU\_simulationDeroulement :

Définition/objectif :

Jeu de test vérifiant qu'il n'y a pas eu de problème en cours de déroulement de la simulation C++.

Méthodes/opérations de vérification :

Lecture/contrôle du fichier de traces : traceEcranTests.res.

\*\*\*\*\*  
Jeu de test JEU\_variablesGraphique :

Définition/objectif : Jeu de test vérifiant les variables.

Méthodes/opérations de vérification :

Comparaison visuelle de graphiques. Les données à vérifier sont :

- Données ModelMaker de "lesEntreesTest" :  
Le graphique Graph4 de modele2.mod, représentant V1 en fonction du temps.
- Données C++ de "lesSortiesTest" :  
Graphique avec en abscisses : t.res, ordonnées : V1.res.  
(voir "Note 1.").

Note 1. : ce graphique peut être tracé à l'écran à l'aide de l'outil "affgraphs.py". "affgraphs.py" est un logiciel d'affichage de graphiques (il propose et gère une interface permettant de configurer/paramétrer des représentations graphiques, puis affiche à l'écran les tracés demandés). Il en a été fait une copie "copie\_affgraphs.py" dans "lesSortiesTest" afin de l'y utiliser ( commande : "python affgraphs.py" ) ; pour plus d'informations (guide d'utilisation, spécification ...), consulter entêtes et commentaires du fichier "affgraphs.py".

\*\*\*\*\*  
Jeu de test JEU\_variablesNumerique :

Définition/objectif : Jeu de test vérifiant les variables.

Méthodes/opérations de vérification :

Comparaison "manuelle" de valeurs numériques (on peut s'aider d'un tableur pour calculer les écarts - cf lesSortiesTest/tablesEcart.xls). Les données à vérifier sont

t :

- Données ModelMaker de "lesEntreesTest" :  
Fichier graph4.txt contenant  
la liste des données : t, V1 pour t=0,2,4,...,198,200.
- Données C++ de "lesSortiesTest" :  
Fichier t.res contenant la liste des valeurs du temps t.  
Fichier V1.res contenant la liste des valeurs de la variable V1.

\*\*\*\*\*

Jeu de test JEU\_fluxNumerique :

Définition/objectif : Jeu de test vérifiant les flux.

Méthodes/opérations de vérification :

Comparaison "manuelle" de valeurs numériques (on peut s'aider d'un tableur pour calculer les écarts - cf lesSortiesTest/tablesEcartis.xls). Les données à vérifier sont :

- Données ModelMaker de "lesEntreesTest" :  
Fichier table2.txt contenant  
la liste des données : t, F1, F2, F3, pour t=0,2,4,...,198,200.
- Données C++ de "lesSortiesTest" :  
Fichier t.res contenant la liste des valeurs du temps t  
Fichiers F1.res, F2.res, F3.res contenant  
les listes des valeurs des flux F1,F2,F3.

\*\*\*\*\*

Jeu de test JEU\_compartimentsGraphique :

Définition/objectif : Jeu de test vérifiant les compartiments.

Méthodes/opérations de vérification :

Comparaison visuelle de graphiques. Les données à vérifier sont :

- Données ModelMaker de "lesEntreesTest" :  
Le graphique Graph6 de modele2.mod représentant  
C1\_source, C2\_reservoir, C3\_reservoir, C3max, C4\_puits, C5\_debordement  
en fonction du temps.
- Données C++ de "lesSortiesTest" :  
Graphiques avec en abscisses : t.res, et en ordonnées : C1\_source.res,  
C2\_reservoir.res, C3\_reservoir.res, C4\_puits.res, C5\_debordement.res.  
(voir "Note 1." plus haut).

\*\*\*\*\*

Jeu de test JEU\_compartimentsNumerique :

Définition/objectif : Jeu de test vérifiant les compartiments.

Méthodes/opérations de vérification :

Comparaison "manuelle" de valeurs numériques (on peut s'aider d'un tableur pour calculer les écarts - cf lesSortiesTest/tablesEcartis.xls). Les données à vérifier sont :

- Données ModelMaker de "lesEntreesTest" :  
Fichier graph6.txt contenant la liste des données :  
t, C1\_source, C2\_reservoir, C3\_reservoir, (C3max), C4\_puits, C5\_debordement  
pour t=0,2,4,...,198,200.
- Données C++ de "lesSortiesTest" :  
Fichier t.res contenant la liste des valeurs du temps t.

Fichiers C1\_source.res, C2\_reservoir.res, C3\_reservoir.res, C4\_puits.res, C5\_debordement.res contenant les listes des valeurs des compartiments C1\_source ... C5\_debordement.

\*\*\*\*\*
\*\*\*\*\*
\*
\* PARTIE RAPPORT DE TEST
\*
\*\*\*\*\*

\*\*\*\*\*
\*
\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
\*
\*
\*\*\*\*\*

\*\*\*\*\* Jeu de test JEU\_simulationDeroulement : \*\*\*\*\*

A la lecture du fichier "traceEcranTests.res", il n'est pas trouvé d'anomalie dans les traces :

La simulation est bien effectuée dans les conditions qui ont été demandées (de t = 0 à 200 s, avec sortie de résultats toutes les 2 s).

Les valeurs lues pour C3max, P1, P2, P3, P4, P5 sont correctes.

Les "resultats sortis" pour t, V1, F1, F2, F3, F4, C1\_source, C2\_reservoir, C3\_reservoir, C4\_puits, C5\_debordement, ne sont pas vérifiés dans ce jeu de test; ils le sont par ailleurs (dans les autres jeux de test).

Conclusion du JEU\_simulationDeroulement : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_variablesGraphique : \*\*\*\*\*

Pour V1 :
On observe/compare les courbes MM (tracées sur support papier par impression de Graph4) et les courbes C++ (tracées à l'écran avec "copie\_affgraphs.py").

Pour V1 :
Il est vérifié que les courbes MM et C++ ont la même forme, présentent les mêmes variations pour - "à vue d'oeil" - les mêmes valeurs.

Conclusion du JEU\_variablesGraphique : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_variablesNumerique : \*\*\*\*\*

On calcule comme écarts max (absolus) entre valeurs C++ ET MM :

Pour V1 : MAX( | ECART ( donnée C++, donnée MM ) | ) = 0,0000120000

On considère que ces écarts sont liés à la précision des calculs.

Conclusion du JEU\_variablesNumerique : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_fluxNumerique : \*\*\*\*\*

On calcule comme écarts max (absolus) entre valeurs C++ ET MM :

Pour F1 : MAX( | ECART ( donnée C++, donnée MM ) | ) = 0,0000294000
Pour F2 : MAX( | ECART ( donnée C++, donnée MM ) | ) = 0,0001755000
Pour F3 : MAX( | ECART ( donnée C++, donnée MM ) | ) = 0,0000556300



XX

\*\*\* SCENARIO DE TEST scnCasDerreur1 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*
Description du test :

Ces tests font partie de la famille des tests portant sur le traitement des cas d'erreurs, ie l'ensemble des scénarios { scnCasDerreur\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnCasDerreur\_i } : cette famille de test porte sur le traitement des cas d'erreurs. Il s'agit de vérifier la détection des erreurs, leur signalement et le comportement que le logiciel adopte dans ces situations d'erreurs. On distingue les erreurs qui se produisent en phase d'initialisation et celles qui se produisent en cours de simulation. La gestion des situations d'erreurs est présentée dans "Cas d'erreurs, traces et codes d'erreurs" dans le fichier he lp\_specification, où est également présenté la phase d'initialisation dans "Initialisation du système" (présentation de initialisation\_phase1 et initialisation\_phase2..). Les codes d'erreurs sont définis dans le fichier genericitePortageMM2CPP.h (constantes "CR\_NOK...").

Les tests du scénario scnCasDerreur1 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnCasDerreur1 portent sur des erreurs survenues en phase d'initialisation initialisation\_phase1.

Informations sur la simulation effectuée en C++ : idem scnInitialisation2.

\*\*\*\*\*
Jeu de test JEU\_InitialisationPhase1\_erreurA :

Définition/objectif : idem JEU\_InitialisationPhase1\_erreurA de scnInitialisation2.

Méthodes/opérations de vérification : idem JEU\_InitialisationPhase1\_erreurA de scnInitialisation2.

\*\*\*\*\*
Jeu de test JEU\_InitialisationPhase1\_erreurB :

Définition/objectif : idem JEU\_InitialisationPhase1\_erreurB de sncInitialisation2.

Méthodes/opérations de vérification : idem JEU\_InitialisationPhase1\_erreurB de sncInitialisation2.

\*\*\*\*\*
Jeu de test JEU\_InitialisationPhase1\_erreurC :

Définition/objectif : idem JEU\_InitialisationPhase1\_erreurC de sncInitialisation2.

Méthodes/opérations de vérification : idem JEU\_InitialisationPhase1\_erreurC de sncInitialisation2.



XX

XX

\*\*\* SCENARIO DE TEST scnCasDerreur2 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\* \*\* \*\* \*\*

\*\*\*\*\*

\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*
Description du test :

Ces tests font partie de la famille des tests portant sur le traitement des cas d'erreurs, ie l'ensemble des scénarios { scnCasDerreur\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnCasDerreur\_i } : voir scnCasDerreur1.

Les tests du scénario scnCasDerreur2 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnCasDerreur2 portent sur des erreurs survenues en phase d'initialisation initialisation\_phase2.

Informations sur la simulation effectuée en C++ : voir ci-dessous au niveau de chaque jeu.

\*\*\*\*\*
Jeu de test JEU\_InitialisationPhase2\_erreur :

Définition/objectif : idem JEU\_InitialisationPhase2\_erreur de scnInitialisation3.

Méthodes/opérations de vérification :

Se placer dans les conditions du scénario scnInitialisation3 et jouer le jeu de test JEU\_InitialisationPhase2\_erreur du scénario scnInitialisation3.

\*\*\*\*\*

\*\*\*\*\*
\*
\* PARTIE RAPPORT DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*

\*
\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :



\*
\*
\*

\*\*\*\*\* Jeu de test JEU\_InitialisationPhase2\_erreur : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationPhase2\_erreur : TEST\_OK.

\*\*\*\*\* CONCLUSION BILAN du test effectué le 17/09/08 : \*\*\*\*\*

\*
\* TEST\_OK,
\*
\* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
\*

\*\*\* Statut du SCENARIO DE TEST scnCasDerreur2 : test à rejouer en non régression.

Le SCENARIO DE TEST scnCasDerreur2 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*

scnCasDerreur2 pointe des jeux de :
- scnInitialisation3.

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnCasDerreur3 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* Raport de test \*\*\* \*\*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*

Description du test :

Ces tests font partie de la famille des tests portant sur le traitement des cas d'erreurs, ie l'ensemble des scénarios { scnCasDerreur\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnCasDerreur\_i } : voir scnCasDerreur1.

Les tests du scénario scnCasDerreur3 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnCasDerreur3 portent sur des erreurs survenues en phase de déroulement de la simulation (ie phase qui suit la phase d'initialisation lorsque celle-ci s'est bien passée).

En fait, le projet de scénario scnCasDerreur3 N'A PAS ETE MIS EN OEUVRE car il n'était pas facile de se placer dans les conditions du test (difficile de provoquer simplement une anomalie du type souhaité).

=> pas de jeu de test pour le scénario scnCasDerreur3, partie rapport de test vide.

\*\*\*\*\*

Jeu de test prévu à l'origine mais NON ECRIT/NON CREE :

Définition/objectif :

Préalable : Description des conséquences pour le logiciel d'une erreur détectée en cours de simulation : se reporter à "Cas d'erreurs, traces et codes d'erreurs" dans le fichier help\_specification ("Erreurs selon les phases du logiciel" / "Erreurs en cours de simulation").

Jeu de test vérifiant que le logiciel détecte l'erreur en cours de traitement et réagit à l'anomalie détectée : cas anormal d'une erreur détectée en cours de simulation. Il s'agit de vérifier que l'anomalie n'a aucune conséquence avant l'instant où elle survient/est décelée, et qu'à partir de l'instant où elle survient/est détectée, les calculs sont arrêtés.

Méthodes/opérations de vérification :

Vérifier que initialisation\_phase1 et initialisation\_phase2 se sont bien déroulés, que la simulation est lancée mais n'est pas déroulée correctement jusqu'à la fin : affichage du message : "\*\*\*\*\* FIN PREMATUREE DE LA SIMULATION à l'instant courant : ..." où l'instant affiché suit l'instant de l'anomalie (message survenant en cas d'interruption de la boucle d'évolution dans le temps). Vérifier que l'erreur considérée est relevée, que dans les fichiers \*.res il manque les valeurs d'après l'instant de l'anomalie.

\*\*\*\*\*

\*\*\*\*\*

\*  
\* PARTIE RAPPORT DE TEST \*  
\*

\*\*\*\*\*

vide

\*\*\*\*\*

\*\*\* Statut du SCENARIO DE TEST scnCasDerreur3 : projet de scénario non mis en oeuvre .

Le SCENARIO DE TEST scnCasDerreur3 a été défini pour vérifier la version 080917 du logiciel.

Le SCENARIO DE TEST scnCasDerreur3 n'a pas été mis en oeuvre car il n'était pas facile de se placer dans les conditions du test (difficile de provoquer simplement une anomalie du type souhaité). Il est gardé en l'état uniquement pour mémoire.

\*\*\*\*\*

(scnCasDerreur3 ne pointe pas de jeux d'autres scénarios)

\*\*\*\*\*  
XX

XX

\*\*\* SCENARIO DE TEST scnInitialisation1 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*  
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*  
\* Copyright INRA, février 2006 \*  
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\*

\*\*\*\*\*  
\* PARTIE DESCRIPTION DE TEST \*  
\* \*\*\*\*\*

\*\*\*\*\*  
Description du test :

Ces tests font partie de la famille des tests portant sur la phase d'initialisation du logiciel, ie l'ensemble des scénarios { scnInitialisation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInitialisation\_i } : cette famille de tests porte sur la phase d'initialisation du logiciel (le logiciel comporte 2 phases consécutives : la phase d'initialisation suivie de la phase de déroulement de la simulation). Il s'agit de vérifier la phase d'initialisation dans le cas où elle se passe bien et en cas d'erreur. La phase d'initialisation est présentée dans le fichier help\_specification dans "Initialisation du système" : présentation de initialisation\_phase1 et initialisation\_phase2, voir en particulier "Pseudo-code du programme principal plus détaillé par rapport aux traces/messages affichés".

Les tests du scénario scnInitialisation1 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInitialisation1 portent sur la phase d'initialisation dans le cas nominal (où l'initialisation se passe bien, sans erreur).

Informations sur la simulation effectuée en C++ :  
Se reporter à scnInterpolation1 car la situation de test du scénario scnInitialisation1 est complètement identique à celle du scénario scnInterpolation1.

\*\*\*\*\*  
Jeu de test JEU\_InitialisationOk :

Définition/objectif :  
Jeu de test vérifiant le cas où la phase d'initialisation se passe bien, se déroule sans erreur.

Méthodes/opérations de vérification :  
Observation/analyse du fichier des traces lesSortiesTest/traceEcranTests.res :  
Tout d'abord vérification du bon déroulement de la phase d'initialisation :  
Vérifier que initialisation\_phase1 s'est bien déroulée :

- ABSENCE d'affichage du message " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME" (message qui apparait en cas d'erreur survenue en initialisation\_phase1).
- affichage du message : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." (message qui arrive à la suite de initialisation\_phase1 si initialisation\_phase1 s'est bien passée).

Vérifier que initialisation\_phase2 s'est bien déroulée :

- ABSENCE d'affichage du message " Fin de fonction simuler en erreur ..." (message qui apparait entre autre en cas d'erreur survenue en initialisation\_phase2).
- affichage du message : "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE ... les conditions de la simulation effectuée ..." (message qui arrive après initialisation\_phase2 si initialisation\_phase2 s'est bien passée et qui plus globalement signifie que suite à la bonne initialisation la simulation est lancée).

Puis vérification du bon déroulement de la simulation qui suit :

- affichage des messages "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur" signalant une fin correcte (la simulation s'est bien déroulée jusqu'à la fin).
- vérifier dans lesSortiesTest que les fichiers résultats associés aux entités (.res) ont été générés.

```
*****
*****
*****
*
*          PARTIE RAPPORT DE TEST          *
*
*****
```

```
*****
*
* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
*
*
```

\*\*\*\*\* Jeu de test JEU\_InitialisationOk : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents) ainsi que la présence des fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationOk : TEST\_OK.

```
***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*
* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
*
```

\*\*\* Statut du SCENARIO DE TEST scnInitialisation1 : test à rejouer en non régression .

Le SCENARIO DE TEST scnInitialisation1 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*

scnInitialisation1 pointe des jeux de :  
- scnInterpolation1.

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnInitialisation2 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*  
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*  
\* Copyright INRA, février 2006 \*  
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\*

\*\*\*\*\*  
\*  
\* PARTIE DESCRIPTION DE TEST \*  
\*  
\*\*\*\*\*

\*\*\*\*\*  
Description du test :

Ces tests font partie de la famille des tests portant sur la phase d'initialisation du logiciel, ie l'ensemble des scénarios { scnInitialisation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInitialisation\_i } : voir scnInitialisation1.

Les tests du scénario scnInitialisation2 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInitialisation2 portent sur des cas d'erreurs survenues en phase d'initialisation initialisation\_phasel.

Informations sur la simulation effectuée en C++ : voir ci-dessous au niveau de chaque jeu.

\*\*\*\*\*  
Jeu de test JEU\_InitialisationPhasel\_erreurA :

Définition/objectif :

Jeu de test vérifiant un cas d'erreur survenue en phase d'initialisation initialisation\_phasel : cas anormal d'un fichier de données à interpoler qui ne contient qu'une seule ligne.

Méthodes/opérations de vérification :

Se placer dans les conditions du scénario scnInterpolation3 et jouer le jeu de test JEU\_formatFichierDonnees du scénario scnInterpolation3.

\*\*\*\*\*  
Jeu de test JEU\_InitialisationPhasel\_erreurB :

Définition/objectif :

Jeu de test vérifiant un cas d'erreur survenue en phase d'initialisation initialisation\_phasel : cas anormal d'un fichier mal configuré (incohérence entre nombre et indice de colonnes).

Méthodes/opérations de vérification :

Se placer dans les conditions du scénario scnInterpolation4 et jouer le jeu de test JEU\_formatFichierDonnees du scénario scnInterpolation4.

\*\*\*\*\*

Jeu de test JEU\_InitialisationPhasel\_erreurC :

Définition/objectif :

Jeu de test vérifiant un cas d'erreur survenue en phase d'initialisation initialisation\_phasel : cas anormal d'un fichier introuvable.

Méthodes/opérations de vérification :

Se placer dans les conditions du scénario scnInterpolation5 et jouer le jeu de test JEU\_formatFichierDonnees du scénario scnInterpolation5.

\*\*\*\*\*

\*\*\*\*\*
\*
\* PARTIE RAPPORT DE TEST
\*
\*\*\*\*\*

\*\*\*\*\*
\*
\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
\*
\*\*\*\*\*

\*\*\*\*\* Jeu de test JEU\_InitialisationPhasel\_erreurA : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationPhasel\_erreurA : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_InitialisationPhasel\_erreurB : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationPhasel\_erreurB : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_InitialisationPhasel\_erreurC : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationPhasel\_erreurC : TEST\_OK.

```

***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*
* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
*
*****

```

```

*** Statut du SCENARIO DE TEST scnInitialisation2 : test à rejouer en non régression
.

```

Le SCENARIO DE TEST scnInitialisation2 a été créé pour vérifier la version 080917 du logiciel.

```

*****

```

```

scnInitialisation2 pointe des jeux de :
- scnInterpolation3
- scnInterpolation4
- scnInterpolation5

```

```

*****

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

*** SCENARIO DE TEST scnInitialisation3 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) ***

```

```

*****
* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++      *
* Copyright INRA, février 2006                                             *
*****

```

```

*** ** * ** * ** * ** *      Rapport de test      *** ** * ** * ** * ** *

```

```

*****
*
*          PARTIE DESCRIPTION DE TEST          *
*
*****

```

```

*****

```

Description du test :

Ces tests font partie de la famille des tests portant sur la phase d'initialisation du logiciel, ie l'ensemble des scénarios { scnInitialisation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInitialisation\_i } : voir scnInitialisation1.

Les tests du scénario scnInitialisation3 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInitialisation3 portent sur des cas d'erreurs survenues en phase d'initialisation initialisation\_phase2.

Informations sur la simulation effectuée en C++ :

Se reporter à scnInterpolation1 car la situation de test du scénario scnInitialisation3 est complètement identique à celle du scénario scnInterpolation1 SAUF que les conditions de simulation sont modifiées pour le test afin de se placer dans la situation incorrecte à vérifier. Cette différence (entre la situation de scnInitialisation3 et celle de scnInterpolation1) change le comportement du logiciel.

- Les conditions de simulation (voir mainTestPortageMM2CPP.cpp, genericiteConfigPortageMM2CPP.h) :

genericiteConfigPortageMM2CPP.h (non retouché) : idem scnInterpolation1.

mainTestPortageMM2CPP.cpp (modifié par rapport à scnInterpolation1 pour provoquer l'anomalie CR\_NOK\_PasSimulNegatif : pas de simulation pasSimulation négatif : la simulation est demandée de l'instant tDebutDeLaSimulation=30 s à l'instant tMaxDeLaSimulation=160 s (=30 s + 130 s), avec un pas de simulation pasSimulation=-10 s.

\*\*\*\*\*

Jeu de test JEU\_InitialisationPhase2\_erreur :

Définition/objectif :

Jeu de test vérifiant la situation où survient en phase d'initialisation initialisation\_phase2 l'erreur suivante : pas de simulation pasSimulation négatif (CR\_NOK\_PasSimulNegatif).

remarque : étant données les signatures des fonctions "simuler(...)" actuelles, il n'est pas possible de provoquer simplement (sans artifice) la situation d'erreur : incohérence entre les instants de début et de fin de simulation (CR\_NOK\_BornesIncohérentes).

Méthodes/opérations de vérification :

Préalable : Description des conséquences pour le logiciel d'une erreur détectée dans la phase d'initialisation initialisation\_phase2 : se reporter à "Cas d'erreurs, traces et codes d'erreurs" dans le fichier help\_specification.

Observation/analyse du fichier des traces lesSortiesTest/traceEcranTests.res :

Vérifier que initialisation\_phase1 s'est bien déroulée :

- ABSENCE d'affichage du message " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVÉ A L'INITIALISATION DU SYSTEME" (message qui apparaît en cas d'erreur survenue en initialisation\_phase1).
- affichage du message : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." (message qui arrive à la suite de initialisation\_phase1 si initialisation\_phase1 s'est bien passée).

Vérifier que initialisation\_phase2 s'est mal déroulée et que l'erreur considérée (pas de simulation pasSimulation négatif) est relevée :

- affichage du message signalant l'erreur détectée "Fin de fonction simuler en erreur" avec le code erreur adéquat (message qui apparaît entre autre en cas d'erreur survenue en initialisation\_phase2).
- vérifier dans ce message que ce code erreur est : CR\_NOK\_PasSimulNegatif=12 (défini dans genericitePortageMM2CPP.h).

Vérifier qu'après l'initialisation\_phase2 le logiciel s'arrête sans que ne soit déroulée de simulation :

- ABSENCE d'affichage du message : "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE" (message qui arrive après initialisation\_phase2 si initialisation\_phase2 s'est bien passée, quand la simulation est lancée).
- et à plus forte raison ABSENCE d'affichage des messages "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur" (signalant une fin correcte de la simulation).
- vérifier dans lesSortiesTest que les fichiers résultats associés aux entités (E.res) n'ont pas été générés.

\*\*\*\*\*

\*\*\*\*\*



```

*
*
*
*****

```

PARTIE RAPPORT DE TEST

```

*****
*
* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
*
*
*

```

\*\*\*\*\* Jeu de test JEU\_InitialisationPhase2\_erreur : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationPhase2\_erreur : TEST\_OK.

```

***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*
* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
*
*****

```

\*\*\* Statut du SCENARIO DE TEST scnInitialisation3 : test à rejouer en non régression .

Le SCENARIO DE TEST scnInitialisation3 a été créé pour vérifier la version 080917 du logiciel.

```

*****
(scenInitialisation3 ne pointe pas de jeux d'autres scénarios)
*****

```

XX

XX

\*\*\* SCENARIO DE TEST scnInitialisation5 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

```

*****
* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++      *
* Copyright INRA, février 2006                                             *
*****

```

\*\*\* \*\* Rapport de test \*\*\* \*\*

```

*****
*
*
*
PARTIE DESCRIPTION DE TEST
*

```

\* \*  
 \*\*\*\*\*  
 \*\*\*\*\*

Description du test :

Ces tests font partie de la famille des tests portant sur la phase d'initialisation du logiciel, ie l'ensemble des scénarios { scnInitialisation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInitialisation\_i } : voir scnInitialisation1.

Les tests du scénario scnInitialisation5 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInitialisation5 portent sur des cas non nominaux mais non bloquants (non considérés comme des cas d'erreurs) gérés en phase d'initialisation initiaisation\_phase2.

Informations sur la simulation effectuée en C++ :  
 Se reporter à scnInterpolation1 car la situation de test du scénario scnInitialisation5 est complètement identique à celle du scénario scnInterpolation1 SAUF que les conditions de simulation sont modifiées pour le test afin de se placer dans la situation non nominale à vérifier. Cette différence (entre la situation de scnInitialisation5 et celle de scnInterpolation1) change en partie le comportement du logiciel (réajustement de pasDeLaSimulation).

- Les conditions de simulation (voir mainTestPortageMM2CPP.cpp, genericiteConfigPortageMM2CPP.h) :

genericiteConfigPortageMM2CPP.h (non retouché) : idem scnInterpolation1.

mainTestPortageMM2CPP.cpp (modifié par rapport à scnInterpolation1 : pour JEU\_InitialisationRetouche il s'agit de provoquer la réaction de réajustement de pasDeLaSimulation ; pour JEU\_InitialisationAvertissement pas de modification nécessaire puis que la situation d'extrapolation apparait déjà dans scnInterpolation1) : la simulation est demandée dans les mêmes conditions que scnInterpolation1 SAUF que pas de simulation pasSimulation=3 s (ce qui provoque la réaction de réajustement du pas de simulation à pasDeLaSimulation=5s sachant par ailleurs que deltat=1 s et PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN=5).

\*\*\*\*\*  
 Jeu de test JEU\_InitialisationRetouche :

Définition/objectif :

Jeu de test vérifiant le cas où des vérifications (en phase d'initialisation initiaisation\_phase2) conduisent à adopter la réaction (non bloquante) de faire des retouches : modification/augmentation de pasDeLaSimulation par rapport à la demande (pour des questions de précision des calculs).

Méthodes/opérations de vérification :

- Vérifier que le réajustement de pasDeLaSimulation est effectué et signalé :
  - affichage du message : "\*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ... pasSimulation=3s ...".
  - affichage du message : "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE ... les conditions de la simulation effectuée ... pasDeLaSimulation=5s ...".
  - affichage du message : "\*\*\*\*\* Remarque : pour des questions de précision des calculs, pasDeLaSimulation a été modifié (augmenté) par rapport à la demande, de telle sorte que dans la durée pasDeLaSimulation, il contienne au moins PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN (=5) fois le pas de temps (plus petit incrément) deltat (=1s)".

Vérifier que le logiciel ne se comporte pas comme dans un cas d'anomalie bloquante : ABSENCE d'affichage des messages " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME" et " Fin de fonction simuler en erreur ...".

Vérifier que la simulation est bien lancée et se déroule correctement jusqu'à la

fin :

- affichage des messages : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." et "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE ... les conditions de la simulation effectuée ...".
- affichage des messages : "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur" (signalant une fin correcte).
- vérifier les résultats de simulation sortis : pour cela jouer le JEU\_calculEntiteInterpolee de scnInterpolation1 (ie effectuer les comparaisons "diff" qui y sont décrites). Remarque : étant donnée "la forme rectiligne" des données à interpoler, le changement de pasDeLaSimulation entre scnInterpolation1 (=10s) et scnInitialisation5 (=5s après réajustement) ne doit pas induire de différences (même légères qui seraient liées à des précisions de calcul) entre les résultats de scnInitialisation5 et ceux de scnInterpolation1 ; par rapport à scnInterpolation1, les fichiers des résultats attendus ont été réécrits ici (cf lesEntreestest/readme.txt) simplement pour tenir compte de l'écart ici de 5 s entre 2 valeurs consécutives au lieu de 10 s dans scnInterpolation1 (cf ../scnInterpolation1/lesEntreestest/readme.txt).

\*\*\*\*\*  
 Jeu de test JEU\_InitialisationAvertissement :

Définition/objectif :

Jeu de test vérifiant le cas où des vérifications (en phase d'initialisation initialisation\_phase2) conduisent à adopter la réaction (non bloquante) de formuler un avertissement : avertissement/signalement qu'étant données les conditions de simulation on risque d'être amené à extrapoler par rapport aux données d'un fichier de données à interpoler.

Méthodes/opérations de vérification : idem JEU\_avertissementIntervalle de scnInterpolation1.

\*\*\*\*\*  
 \*\*\*\*\*  
 \*  
 \*  
 \* PARTIE RAPPORT DE TEST \*  
 \*  
 \*\*\*\*\*

\*\*\*\*\*  
 \*  
 \* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :  
 \*  
 \*  
 \*

\*\*\*\*\* Jeu de test JEU\_InitialisationRetouche : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), en particulier le message signalant le réajustement du pas de la simulation.

Pour tous les fichiers comparés 2 à 2, la 1ère valeur diffère systématiquement, de la même manière que dans le jeu JEU\_calculEntiteInterpolee de scnInterpolation1. La différence sur la 1ère valeur ne peut pas être considérée comme une erreur, elle est due au décalage dans l'instant d'affichage initial, lié au démarrage (1er instant d'affichage = tDebutDeLaSimulation + deltat et non pas exactement tDebutDeLaSimulation). Pour plus de précisions, voir le rapport du jeu JEU\_calculEntiteInterpolee de scnInterpolation1. A part cette 1ère valeur, les commandes "diff" ne relèvent aucune différence entre les fichiers comparés.

Conclusion du JEU\_InitialisationRetouche : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_InitialisationAvertissement : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), en particulier l'avertissement attendu ainsi que le comportement attendu (fin de simulation sans erreur) et la présence des fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_InitialisationAvertissement : TEST\_OK.

\*\*\*\*\* CONCLUSION BILAN du test effectué le 17/09/08 : \*\*\*\*\*
\*
\* TEST\_OK,
\*
\* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
\*

\*\*\* Statut du SCENARIO DE TEST scnInitialisation5 : test à rejouer en non régression .

Le SCENARIO DE TEST scnInitialisation5 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*
(scenInitialisation5 ne pointe pas de jeux d'autres scénarios)

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnIntegrale : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

Description du test :

Les vérifications du scénario scnIntegrale portent sur le traitement des entités de type entityAvecDerivee.

Les tests du scénario scnIntegrale ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnIntegrale concernent des vérifications dans une situation nominale.

Informations sur la simulation effectuée en C++ :

Pour plus d'informations que ce qui est décrit ci-dessous, voir directement les fichiers : mainTestPortageMM2CPP.cpp, genericiteConfigPortageMM2CPP.h et specificiteConfigPortageMM2CPP.h

- Les conditions de simulation (voir mainTestPortageMM2CPP.cpp, genericiteConfigPortageMM2CPP.h) :

Simulation de l'instant tDebutDeLaSimulation=40 s à l'instant tMaxDeLaSimulation=160 s, avec un pas de simulation pasSimulation=1 s, et un pas de temps d'horloge laissé à  $\text{deltat}=0.001$  s (la configuration dans le fichier genericiteConfigPortageMM2CPP.h n'est pas retouchée pour le test). Remarque : par rapport à  $\text{deltat}$  et à PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN (qui vaut 1000, cf fichier genericiteConfigPortageMM2CPP.h), le pas de simulation pasSimulation 1 s (=  $0.001$  s x 1000) est ici minimal (on est dans la situation avec la précision de calcul la plus faible).

- Les entités de type entityAvecDerivee considérées :

Le scénario s'appuie sur 4 entités de type entityAvecDerivee : les compartiments (compartiment) C\_constante, C\_monte, C\_descend, C\_sinusoidale (compartiments relatifs à l'ajout m\_integrale).

Chaque compartiment a pour valeur (value) à l'instant courant (variant de  $t=40$  s à  $t=160$  s) : l'intégrale de sa ddt en fonction du temps entre  $t=40$  s et l'instant courant. Pour plus de renseignements, voir "lesEntreesTest/readme".

Compartiment C\_constante :

- sa valeur "initiale" (ie à  $t=40$ s) est 7.0 et
- sa fonction dérivée ddt est :  $\text{ddt} = 50$

Compartiment C\_monte :

- sa valeur "initiale" (ie à  $t=40$ s) est 0.0 et
- sa fonction dérivée ddt est :  $\text{ddt} = 200 + 30 t$  (droite passant par les points ( $t=40, 1400$ ) et ( $t=160, 5000$ ))

Compartiment C\_descend :

- sa valeur "initiale" (ie à  $t=40$ s) est 0.0 et
- sa fonction dérivée ddt est :  $\text{ddt} = 6200 - 30 t$  (droite passant par les points ( $t=40, 5000$ ) et ( $t=160, 1400$ ))

Compartiment C\_sinusoidale :

- sa valeur "initiale" (ie à  $t=40$ s) est 0.0 et
- sa fonction dérivée ddt est :  $\text{ddt} = \sin( t \times \text{PI}/20 - 2\text{PI} )$

- Les données résultats : voir "lesSortiesTest"

\*\*\*\*\*  
Jeu de test JEU\_calculValueGraphique :

Définition/objectif :

Jeu de test vérifiant à un premier niveau ("simple" contrôle visuel) le calcul de 4 valeurs d'entités de type entityAvecDerivee : value des 4 compartiments (compartiment) C\_constante, C\_monte, C\_descend, C\_sinusoidale calculées entre  $t=40$  s et  $t=160$  s

Méthodes/opérations de vérification :

Commencer par vérifier que t\_attendu.res et t.res coïncident à  $\text{deltat}$  près. Pour cela, commande diff :

```
diff lesEntreesTest/t_attendu.res lesSortiesTest/t.res
```

Puis comparaison visuelle de graphiques (voir "Note 1."). En plaçant en abscisses de tous les graphiques les données lesEntreesTest/t\_attendu.res, les données à placer en ordonnées pour vérifications/comparaisons sont :

- Données attendues (précalculées) de "lesEntreesTest" : C\_constante\_attendu.res, C\_monte\_attendu.res, C\_descend\_attendu.res, C\_sinusoidale\_attendu.res, qui sont les résultats du test attendus pour les compartiments respectifs C\_constante, C\_monte, C\_descend, C\_sinusoidale, à comparer respectivement avec
- Données C++ de "lesSortiesTest" : C\_constante.res, C\_monte.res, C\_descend.res, C\_sinusoidale.res

sinusoidale.res.

Note 1. : les graphiques peuvent être tracés à l'écran à l'aide de l'outil "affgraphs.py". "affgraphs.py" est un logiciel d'affichage de graphiques (il propose et gère une interface permettant de configurer/paramétrer des représentations graphiques, puis affiche à l'écran les tracés demandés). Il en a été fait une copie "copie\_affgraphs.py" dans "lesSortiesTest" afin de l'y utiliser ( commande : "python affgraphs.py" ) ; pour plus d'informations (guide d'utilisation, spécification ...), consulter en têtes et commentaires du fichier "affgraphs.py".

\*\*\*\*\*

Jeu de test JEU\_calculValueNumerique :

Définition/objectif :

Jeu de test vérifiant le calcul des valeurs d'entités de type entityAvecDerivee : value des 4 compartiments (compartiment) C\_constante, C\_monte, C\_descend, C\_sinusoidale calculées entre t=40 s et t=160 s.

Méthodes/opérations de vérification :

Commencer par vérifier que t\_attendu.res et t.res coïncident à deltat près. Pour cela, commande diff :

```
diff lesEntreesTest/t_attendu.res lesSortiesTest/t.res
```

Puis comparaison 2 à 2 des fichiers relatifs aux compartiments C\_constante, C\_monte, C\_descend, C\_sinusoidale. Les données à comparer sont :

- Données C++ de "lesSortiesTest" : C\_constante.res, C\_monte.res, C\_descend.res, C\_sinusoidale.res, et

- Données de "lesEntreesTest" précalculées pour le test : C\_constante\_attendu.res, C\_monte\_attendu.res, C\_descend\_attendu.res, C\_sinusoidale\_attendu.res.

1) Vérification globale : pour chaque comparaison de fichiers 2 à 2 (C\_xxx.res et C\_xxx\_attendu.res), il s'agit de vérifier que l'écart observé entre les valeurs (valeur de C\_xxx et valeur de C\_xxx\_attendu à chaque instant) reste dû à (reste dans la limite de) la précision des calculs. Cependant l'écart alors observé peut en partie venir d'un décalage existant entre t et t\_attendu. D'où une vérification plus poussée :

2) Vérification fine : même vérification que ci-dessus ("1) Vérification globale") SAUF qu'on ne tient compte que des couples (valeur de C\_xxx, valeur de C\_xxx\_attendu) relatifs aux instants tels que t et t\_attendu coïncident exactement.

La comparaison peut être faite par commande "diff" :

```
diff lesEntreesTest/C_constante_attendu.res lesSortiesTest/C_constante.res
diff lesEntreesTest/C_monte_attendu.res lesSortiesTest/C_monte.res
diff lesEntreesTest/C_descend_attendu.res lesSortiesTest/C_descend.res
diff lesEntreesTest/C_sinusoidale_attendu.res lesSortiesTest/C_sinusoidale.res
```

Ou en s'aidant d'un tableur pour calculer les écarts (cf lesSortiesTest/tablesEcart.xls).

\*\*\*\*\*

\*\*\*\*\*

\*
\*
\* PARTIE RAPPORT DE TEST
\*
\*
\*\*\*\*\*

\*\*\*\*\*

\*
\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
\*
\*
\*

\*\*\*\*\* Jeu de test JEU\_calculValueGraphique : \*\*\*\*\*

La commande "diff lesEntreesTest/t\_attendu.res lesSortiesTest/t.res" ne relève pas de différence supérieure à  $\Delta t$  entre les fichiers comparés. Plus précisément il y a exacte égalité entre les valeurs comparées, sauf pour la 1ère valeur sur laquelle il est trouvé une différence de  $0.001s = \Delta t$  (40.000999 s au lieu de 40.000000 s attendu). Pour plus de précisions sur ce décalage dans l'instant d'affichage initial, voir le rapport du jeu JEU\_calculEntiteInterpolee de scnInterpolation1.

Pour C\_xxx = C\_constante, C\_monte, C\_descend, C\_sinusoidale : il est vérifié que les courbes lesSortiesTest/C\_xxx.res et lesEntreesTest/C\_xxx\_attendu.res (tracées pour observation/comparaison à l'écran avec "copie\_affgraphs.py") ont la même forme, présentent les mêmes variations pour - "à vue d'oeil" - les mêmes valeurs.

Conclusion du JEU\_calculValueGraphique : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_calculValueNumerique : \*\*\*\*\*

La commande "diff lesEntreesTest/t\_attendu.res lesSortiesTest/t.res" : voir jeu JEU\_calculValueGraphique ci-dessus.

1) Vérification globale (rappel :  $\Delta t = 0,001$  s) :

Contrairement à ce qui est demandé, ce ne sont pas tous les couples (valeur de C\_xxx, valeur de C\_xxx\_attendu) qui sont pris en compte : on exclut d'office le 1er couple pour lequel on sait que systématiquement  $t_{\text{attendu}} < t$  (et même  $t = t_{\text{attendu}} + \Delta t$ ).

On calcule comme écarts max (absolus) entre valeurs issues des calculs/traitements (fichiers lesSortiesTest/C\_xxx.res) et valeurs attendues/précalculées pour test (fichiers lesEntreesTest/C\_xxx\_attendu.res) :

Pour C\_constante :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000000  
 Pour C\_monte :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 1,812500  
 Pour C\_descend :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 1,812500  
 Pour C\_sinusoidale :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000502

On considère que le niveau de précision des calculs justifie ces écarts (sans regarder dans le détail l'influence des décalages entre  $t$  et  $t_{\text{attendu}}$ ).

2) Vérification fine (seuls sont pris en compte les couples (valeur de C\_xxx, valeur de C\_xxx\_attendu) relatifs aux instants tels que  $t$  et  $t_{\text{attendu}}$  coïncident exactement) :

On calcule comme écarts max (absolus) entre valeurs issues des calculs/traitements (fichiers lesSortiesTest/C\_xxx.res) et valeurs attendues/précalculées pour test (fichiers lesEntreesTest/C\_xxx\_attendu.res) :

Pour C\_constante :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000000  
 Pour C\_monte :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 1,812500  
 Pour C\_descend :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 1,812500  
 Pour C\_sinusoidale :  
 MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000502

On considère que ces écarts sont liés à la précision des calculs.

Ces écarts reflètent fidèlement la précision des calculs (dans des cas précis : droite, sinusoïde).

Conclusion du JEU\_calculValueNumerique : TEST\_OK.

```

***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*
* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
*
*****

```

```

*** Statut du SCENARIO DE TEST scnIntegrale : test à rejouer en non régression.
Le SCENARIO DE TEST scnIntegrale a été créé pour vérifier la version 080917 du logiciel.

```

```

*****
(scenIntegrale ne pointe pas de jeux d'autres scénarios)
*****

```

XX

XX

```

*** SCENARIO DE TEST scnInterpolation1 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) ***

```

```

*****
* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++      *
* Copyright INRA, février 2006                                           *
*****

```

```

*** ** Report de test *** **

```

```

*****
*
* PARTIE DESCRIPTION DE TEST
*
*****

```

```

*****
Description du test :

```

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : cette famille de tests porte sur le traitement d'interpolation. Elle se rapporte à la modification m\_linearInterpolation. Elle concerne les classes entityInterpoelee et controlled, fichierDentree et donneesAinterpoler. Il s'agit de vérifier (i) que les fichiers contenant les données à interpoler sont bien gérés, (ii) que les entités interpolées sont bien



n calculées à partir de ceux-ci, et ensuite (iii) l'utilisation des entités interpolées dans le système par les autres entités (flux...).

Les tests du scénario scnInterpolation1 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation1 concernent des vérifications dans une situation nominale.

Informations sur la simulation effectuée en C++ :

Pour plus d'informations que ce qui est décrit ci-dessous, voir directement les fichiers : mainTestPortageMM2CPP.cpp, genericiteConfigPortageMM2CPP.h et specificiteConfigPortageMM2CPP.h

- Les conditions de simulation (voir mainTestPortageMM2CPP.cpp, genericiteConfigPortageMM2CPP.h) :

Simulation de l'instant tDebutDeLaSimulation=30 s à l'instant tMaxDeLaSimulation=160 s, avec un pas de simulation pasSimulation=10 s, et un pas de temps d'horloge forcé à  $\Delta t = 1$  s (cf configuration dans le fichier genericiteConfigPortageMM2CPP.h réécrit pour le test).

- Les fichiers des données à interpoler (voir specificiteConfigPortageMM2CPP.h, lesEntreesCPP) :

Le scénario s'appuie sur 3 entités interpolées (I1\_pluie, I20, I25) qui correspondent à la même donnée réelle. Cette donnée réelle suit la fonction suivante (une demi-droite ascendante passant par (10,500) et une demi-droite descendante passant par (190,500) qui concourent au point (70,3500)) : pour  $10 \leq t \leq 70$  Valeur =  $50 \times t$ , pour  $70 \leq t \leq 190$  Valeur =  $-25 \times t + 5250$ .

Les 3 entités interpolées (I1\_pluie, I20, I25) sont lues dans des colonnes distinctes, ne proviennent pas toutes du même fichier (voir constantes nomFichierIn\_DONNEE... dans specificiteConfigPortageMM2CPP.h), ne reposent pas toutes sur une seule même liste d'instants. Leurs listes de valeurs et listes d'instants sont "encadrées par les points" (t=10, V=500) et (t=190, V=500). Les instants des données à interpoler lues ne tombent pas tous à des instants du pas de la simulation.

Le scénario s'appuie aussi sur 3 flux (Fpluie1, Fpluie20, Fpluie25) calculés à partir des entités interpolées (respectivement I1\_pluie, I20, I25) selon une seule même règle de calcul  $y = -x / 10.0$ .

- Les données résultats : voir "lesSortiesTest"

\*\*\*\*\*  
Jeu de test JEU\_calculEntiteInterpolee :

Définition/objectif :

Jeu de test vérifiant que les entités interpolées sont bien calculées à partir des fichiers contenant les données à interpoler, et vérifiant l'utilisation des entités interpolées dans le système par les autres entités (flux...).

Méthodes/opérations de vérification :

Comparaison par commande "diff". Les données à vérifier sont :

- Données C++ de "lesSortiesTest" : les données relatives à t, I1\_pluie, I20, I25, Fpluie1, Fpluie20, Fpluie25 (correspondant aux fichiers définis dans specificiteConfigPortageMM2CPP.h par nomFichierOut\_t, nomFichierOut\_I1\_pluie,...).

- à comparer aux données précalculées pour le test :

"lesEntreesTest/t\_attendu.res" les résultats du test attendus pour t.

"lesEntreesTest/I\_attendu.res" les résultats du test attendus pour l'entité interpolée (I1\_pluie, I20, I25).

"lesEntreesTest/F\_attendu.res" les résultats du test attendus pour le flux (Fpluie

e1, Fpluie20, Fpluie25).

Les commandes de vérification à lancer, qui à chaque fois doivent montrer que les 2 fichiers comparés sont complètement identiques, sont (à adapter si jamais les noms des fichiers changent) :

```
diff lesEntreesTest/t_attendu.res lesSortiesTest/t.res
diff lesEntreesTest/I_attendu.res lesSortiesTest/I1_pluie.res
diff lesEntreesTest/I_attendu.res lesSortiesTest/I20.res
diff lesEntreesTest/I_attendu.res lesSortiesTest/I25.res
diff lesEntreesTest/F_attendu.res lesSortiesTest/Fpluie1.res
diff lesEntreesTest/F_attendu.res lesSortiesTest/Fpluie20.res
diff lesEntreesTest/F_attendu.res lesSortiesTest/Fpluie25.res
```

\*\*\*\*\*  
Jeu de test JEU\_avertissementIntervalle :

Définition/objectif :

Jeu de test vérifiant la détection d'un cas où l'intervalle de simulation [ tDebutDeLaSimulation, tMaxDeLaSimulation ] dépasse le domaine des données à interpoler lues dans le fichier. Cette situation conduit à extrapoler lors des calculs d'interpolation, ce qui est signalé à l'utilisateur au titre d'avertissement mais n'est pas considéré comme une anomalie bloquante.

Méthodes/opérations de vérification :

Observation/analyse du fichier des traces lesSortiesTest/traceEcranTests.res :

Vérifier que le logiciel ne s'arrête pas comme dans le cas d'une anomalie bloquante : ABSENCE d'affichage des messages " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVÉ A L'INITIALISATION DU SYSTEME" et " Fin de fonction simuler en erreur ...".

Vérifier que les contrôles en début de simulation ont relevé que le calcul de l'entité interpolée I25 comprendra de l'extrapolation : message ("\*\*\*\*\* AVERTISSEMENT en conclusion de la comparaison des conditions de simulation...") d'avertissement d'extrapolation en début de simulation, car l'intervalle de simulation [ tDebutDeLaSimulation=30s, tMaxDeLaSimulation=160s ] dépasse les bornes [ t=55s, t=115s ] de l'intervalle de données à interpoler lues pour I25 (première donnée à interpoler lue (t=55, V=2750), dernière donnée à interpoler lue (t=115, V=2375)).

Vérifier que malgré cet avertissement la simulation est bien lancée et se déroule correctement jusqu'à la fin :

- affichage des messages : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." et "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE ... les conditions de la simulation effectuée ...".
- affichage des messages : "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur" (signalant une fin correcte).
- vérifier dans lesSortiesTest que les fichiers résultats associés aux entités (.res) ont été générés.

\*\*\*\*\*  
\*\*\*\*\*  
\*  
\*  
\* PARTIE RAPPORT DE TEST \*  
\*  
\*\*\*\*\*

\*\*\*\*\*  
\*  
\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :  
\*  
\*  
\*

\*\*\*\*\* Jeu de test JEU\_calculEntiteInterpolee : \*\*\*\*\*

Les commandes "diff" ne relèvent aucune différence entre les fichiers comparés, mise à part la lère valeur qui diffère systématiquement. La lère valeur trouvée est :

31.000000 pour lesSortiesTest/t.res  
 au lieu de 30.000000 attendu (cf lesEntreesTest/t\_attendu.res)

1550.000000 pour lesSortiesTest/I1\_pluie.res  
 1550.000000 pour lesSortiesTest/I20.res  
 1550.000000 pour lesSortiesTest/I25.res  
 au lieu de 1500.000000 (cf lesEntreesTest/I\_attendu.res)

-155.000000 pour lesSortiesTest/Fpluie1.res  
 -155.000000 pour lesSortiesTest/Fpluie20.res  
 -155.000000 pour lesSortiesTest/Fpluie25.res  
 au lieu de -150.000000 (cf lesEntreesTest/F\_attendu.res)

La différence observée s'explique par le fait que le 1er instant d'affichage (cf t.res et t\_attendu.res) est  $t=31s$  ( $= tDebutDeLaSimulation(=30s) + \text{deltat}(=1s)$ ) et non pas exactement  $t=30s$  ( $= tDebutDeLaSimulation$ ). Les lères valeurs trouvées (1550.000000 pour I1\_pluie.res et I20.res et I25.res, -155.000000 pour Fpluie1.res et Fpluie20.res et Fpluie25.res) sont correctes pour  $t=31$  (voir lesEntreesTest/readme), tandis que celles des fichiers I\_attendu.res et F\_attendu.res (1500.000000 et -150.000000) ont été calculées pour  $t=30$ .

La différence sur la lère valeur ne peut pas être considérée comme une erreur, elle est due au décalage dans l'instant d'affichage initial, lié au démarrage (1er instant d'affichage =  $tDebutDeLaSimulation + \text{deltat}$  et non pas exactement  $tDebutDeLaSimulation$ ).

Conclusion du JEU\_calculEntiteInterpolee : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_avertissementIntervalle : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), en particulier l'avertissement attendu (on sera amené à extrapoler pour le fichier d'entités à interpoler ../lesEntreesCPP/donneesPluieDouble.txt associé à I25, dont l'intervalle de temps des données à interpoler est [ 55, 115 ] alors que  $[tDebutDeLaSimulation, tMaxDeLaSimulation]=[ 30, 160 ]$ ) ainsi que le comportement attendu (fin de simulation sans erreur) et la présence des fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_avertissementIntervalle : TEST\_OK.

\*\*\*\*\* CONCLUSION BILAN du test effectué le 17/09/08 : \*\*\*\*\*

- \* TEST\_OK,
- \* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.

\*\*\*\*\*

\*\*\* Statut du SCENARIO DE TEST scnInterpolation1 : test à rejouer en non régression.

Le SCENARIO DE TEST scnInterpolation1 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*

(scnInterpolation1 ne pointe pas de jeux d'autres scénarios)

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnInterpolation2 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*
Description du test :

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : voir scnInterpolation1.

Les tests du scénario scnInterpolation2 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation2 concernent des vérifications dans une situation nominale.

Informations sur la simulation effectuée en C++ :
Se reporter à scnInterpolation1 car la situation de test du scénario scnInterpolation2 est complètement identique à celle du scénario scnInterpolation1 SAUF qu'ici la configuration du code source testé n'est pas modifiée : pas de réécriture de genericiteConfigPortageMM2CPP.h, le pas de temps d'horloge deltat est celui du code source testé. La différence (entre la situation de scnInterpolation2 et celle de scnInterpolation1) ne doit rien changer aux résultats (à partir du moment où le code source testé ne provoque pas un réajustement de pasDeLaSimulation).

\*\*\*\*\*
Jeu de test JEU\_calculEntiteInterpolee :

Définition/objectif : idem JEU\_calculEntiteInterpolee de scnInterpolation1.

Méthodes/opérations de vérification : idem JEU\_calculEntiteInterpolee de scnInterpolation1.

\*\*\*\*\*
Jeu de test JEU\_avertissementIntervalle :

Définition/objectif : idem JEU\_avertissementIntervalle de scnInterpolation1.

Méthodes/opérations de vérification : idem JEU\_avertissementIntervalle de scnInterpolation1.

```

*****
*****
*
*
*
*
*****

```

PARTIE RAPPORT DE TEST

```

*****
*
*
*
*

```

RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :

\*\*\*\* Jeu de test Jeu de test JEU\_calculEntiteInterpolee : \*\*\*\*

Les commandes "diff" ne relèvent aucune différence entre les fichiers comparés, mise à part la lère valeur qui diffère systématiquement. La lère valeur trouvée est :

30.000999 pour lesSortiesTest/t.res au lieu de 30.000000 attendu (cf lesEntreesTest/t\_attendu.res)

1500.050049 pour lesSortiesTest/I1\_pluie.res
1500.050049 pour lesSortiesTest/I20.res
1500.050049 pour lesSortiesTest/I25.res
au lieu de 1500.000000 (cf lesEntreesTest/I\_attendu.res)

-150.005005 pour lesSortiesTest/Fpluie1.res
-150.005005 pour lesSortiesTest/Fpluie20.res
-150.005005 pour lesSortiesTest/Fpluie25.res
au lieu de -150.000000 (cf lesEntreesTest/F\_attendu.res)

La différence observée s'explique par le fait que le 1er instant d'affichage (cf t.r es et t\_attendu.res) est t=30.000999s (= tDebutDeLaSimulation(=30s) + deltat(=0,001s )) et non pas exactement t=30s (= tDebutDeLaSimulation). Les lères valeurs trouvées (1500.050049 pour I1\_pluie.res et I20.res et I25.res, -150.005005 pour Fpluie1.res e t Fpluie20.res et Fpluie25.res) sont correctes pour t=30,001 (voir lesEntreesTest/re adme), tandis que celles des fichiers I\_attendu.res et F\_attendu.res (1500.000000 et -150.000000) ont été calculées pour t=30.

La différence sur la lère valeur ne peut pas être considérée comme une erreur, elle est due au décalage dans l'instant d'affichage initial, lié au démarrage (1er instan t d'affichage = tDebutDeLaSimulation + deltat et non pas exactement tDebutDeLaSimula tion).

Conclusion du JEU\_calculEntiteInterpolee : TEST\_OK.

\*\*\*\* Jeu de test JEU\_avertissementIntervalle : \*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), en particulier l'avertissement a ttendu (on sera amené à extrapoler pour le fichier d'entités à interpoler ../lesEntr eesCPP/donneesPluieDouble.txt associé à I25, dont l'intervalle de temps des données à interpoler est [ 55, 115 ] alors que [tDebutDeLaSimulation,tMaxDeLaSimulation]=[ 3 0, 160 ]) ainsi que le comportement attendu (fin de simulation sans erreur) et la pr ésence des fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_avertissementIntervalle : TEST\_OK.

```

***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*

```

\* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.

\*
\*\*\*\*\*

\*\*\* Statut du SCENARIO DE TEST scnInterpolation2 : test à rejouer en non régression.

Le SCENARIO DE TEST scnInterpolation2 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*

(scnInterpolation2 ne pointe pas de jeux d'autres scénarios)

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnInterpolation3 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* Report de test \*\*\* \*\* \*\* \*\* \*\*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*
Description du test :

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : voir scnInterpolation1.

Les tests du scénario scnInterpolation3 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation3 concernent des vérifications dans une situation non nominale.

Informations sur la simulation effectuée en C++ :
Se reporter à scnInterpolation1 car la situation de test du scénario scnInterpolation3 est complètement identique à celle du scénario scnInterpolation1 SAUF que les fichiers des données à interpoler sont modifiés pour le test afin de se placer dans la situation incorrecte à vérifier. Cette différence (entre la situation de scnInterpolation3 et celle de scnInterpolation1) change le comportement du logiciel.

- Les fichiers des données à interpoler (voir specificiteConfigPortageMM2CPP.h) :
specificiteConfigPortageMM2CPP.h (configuration des fichiers non retouchée) : i

dem scnInterpolation1.

Le fichier lesEntreesCPP/donneesPluie.txt (contenu modifié) : présente l'anomalie de ne contenir qu'une seule ligne.

Le fichier lesEntreesCPP/donneesPluieDouble.txt : idem scnInterpolation1.

\*\*\*\*\*

Jeu de test JEU\_formatFichierDonnees :

Définition/objectif :

Jeu de test vérifiant que les fichiers contenant les données à interpoler sont bien gérées : cas anormal d'un fichier qui ne contient qu'une seule ligne. Il s'agit d'une erreur détectée dans la phase d'initialisation initialisation\_phasel (présentée dans le fichier help\_specification dans "Initialisation du système").

Méthodes/opérations de vérification :

Préalable : Description des conséquences pour le logiciel d'une erreur détectée dans la phase d'initialisation initialisation\_phasel : se reporter à "Cas d'erreurs, traces et codes d'erreurs" dans le fichier help\_specification.

Observation/analyse du fichier des traces lesSortiesTest/traceEcranTests.res :

Vérifier que initialisation\_phasel s'est mal déroulée et que l'erreur considérée (le fait que le fichier ne contient qu'une seule ligne) est relevée :

- vérifier dans traceEcranTests.res l'affichage du message signalant l'erreur détectée " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME" avec le code erreur adéquat (message qui apparaît en cas d'erreur survenue en initialisation\_phasel).

- vérifier dans ce message que ce code erreur est : CR\_NOK\_FichierInvalide=14 (défini dans genericitePortageMM2CPP.h).

- vérifier dans traceEcranTests.res l'affichage en plus (avant) du message de précision/explication concernant l'erreur rencontrée (le message n'apparaît que si la clé de compilation CLE\_AVEC\_TRACE\_ECRAN\_DONNEES est activée) : " \*\*\*\*\* Fichier d'entités à interpoler : non valide" pour le fichier lesEntreesCPP/donneesPluie.txt.

Vérifier qu'après l'initialisation\_phasel, le logiciel s'arrête sans que ne soit déroulée de simulation :

- ABSENCE d'affichage du message : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." (message qui n'arrive à la suite de initialisation\_phasel que si initialisation\_phasel s'est bien passée).

- et à plus forte raison ABSENCE d'affichage des messages : "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE" et "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur".

- vérifier dans lesSortiesTest que les fichiers résultats associés aux entités (.res) n'ont pas été générés.

\*\*\*\*\*

\*\*\*\*\*

\*

\* PARTIE RAPPORT DE TEST \*

\*

\*\*\*\*\*

\*\*\*\*\*

\*

\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :

\*

\*

\*

\*\*\*\*\* Jeu de test JEU\_formatFichierDonnees : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_formatFichierDonnees : TEST\_OK.

\*\*\*\*\* CONCLUSION BILAN du test effectué le 17/09/08 : \*\*\*\*\*
\*
\* TEST\_OK,
\*
\* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
\*
\*\*\*\*\*

\*\*\* Statut du SCENARIO DE TEST scnInterpolation3 : test à rejouer en non régression.

Le SCENARIO DE TEST scnInterpolation3 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*
(scenInterpolation3 ne pointe pas de jeux d'autres scénarios)
\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnInterpolation4 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*
\* Copyright INRA, février 2006 \*
\*\*\*\*\*

\*\*\* \*\* \* \*\* \* \*\* \* \*\* \* Rapport de test \*\*\* \*\* \* \*\* \* \*\* \* \*\* \* \*\* \*

\*\*\*\*\*
\*
\* PARTIE DESCRIPTION DE TEST \*
\*
\*\*\*\*\*

\*\*\*\*\*
Description du test :

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : voir scnInterpolation1.

Les tests du scénario scnInterpolation4 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation4 concernent des vérifications dans une situat



ion non nominale.

Informations sur la simulation effectuée en C++ :

Se reporter à `scnInterpolation1` car la situation de test du scénario `scnInterpolation4` est complètement identique à celle du scénario `scnInterpolation1` SAUF que les fichiers des données à interpoler sont modifiés pour le test afin de se placer dans la situation incorrecte à vérifier. Cette différence (entre la situation de `scnInterpolation4` et celle de `scnInterpolation1`) change le comportement du logiciel.

- Les fichiers des données à interpoler (voir `specificiteConfigPortageMM2CPP.h`) :

`specificiteConfigPortageMM2CPP.h` (configuration des fichiers retouchée) :

Pour l'entité interpolée `I1_pluie` associée au fichier `lesEntreesCPP/donneesPluie.txt`, `nbColonnes_DONNEE_I1_pluie` est mis à 1 (au lieu de sa valeur correcte : 2), ce qui provoque une incohérence avec l'indice de colonne `colValeur_DONNEE_I1_pluie=1`.

Le fichier `lesEntreesCPP/donneesPluie.txt` : idem `scnInterpolation1`.

Le fichier `lesEntreesCPP/donneesPluieDouble.txt` : idem `scnInterpolation1`.

\*\*\*\*\*  
Jeu de test `JEU_formatFichierDonnees` :

Définition/objectif :

Jeu de test vérifiant que les fichiers contenant les données à interpoler sont bien gérées : cas anormal d'un fichier mal configuré (incohérence entre nombre et indice de colonnes). Il s'agit d'une erreur détectée dans la phase d'initialisation `initialisation_phasel` (présentée dans le fichier `help_specification` dans "Initialisation du système").

Méthodes/opérations de vérification :

Préalable : Description des conséquences pour le logiciel d'une erreur détectée dans la phase d'initialisation `initialisation_phasel` : se reporter à "Cas d'erreurs, traces et codes d'erreurs" dans le fichier `help_specification`.

Observation/analyse du fichier des traces `lesSortiesTest/traceEcranTests.res` :

Vérifier que `initialisation_phasel` s'est mal déroulée et que l'erreur considérée (le fait que le fichier est mal configuré : incohérence entre nombre et indice de colonnes) est relevée :

- vérifier dans `traceEcranTests.res` l'affichage du message signalant l'erreur détectée " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME" avec le code erreur adéquat (message qui apparaît en cas d'erreur survenue en `initialisation_phasel`).

- vérifier dans ce message que ce code erreur est : `CR_NOK_FichierInvalide=14` (défini dans `genericitePortageMM2CPP.h`).

- vérifier dans `traceEcranTests.res` l'affichage en plus (avant) du message de précision/explication concernant l'erreur rencontrée (le message n'apparaît que si la clé de compilation `CLE_AVEC_TRACE_ECRAN_DONNEES` est activée) : " \*\*\*\*\* Fichier d'entités à interpoler : incohérence entre le nombre de colonnes et les indices de colonnes" pour le fichier `lesEntreesCPP/donneesPluie.txt`.

- vérifier dans `traceEcranTests.res` l'affichage en plus (avant) du message de précision/explication concernant l'erreur rencontrée (le message n'apparaît que si la clé de compilation `CLE_AVEC_TRACE_ECRAN_DONNEES` est activée) : " \*\*\*\*\* Fichier d'entités à interpoler : non valide" pour le fichier `lesEntreesCPP/donneesPluie.txt`.

Vérifier qu'après l'initialisation `initialisation_phasel`, le logiciel s'arrête sans que ne soit déroulée de simulation :

- ABSENCE d'affichage du message : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." (message qui n'arrive à la suite de `initialisation_phasel` que si `initialisation_phasel` s'est bien passée).

- et à plus forte raison ABSENCE d'affichage des messages : "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE" et "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur".

- vérifier dans `lesSortiesTest` que les fichiers résultats associés aux entités (`E.res`) n'ont pas été générés.

\*\*\*\*\*

\*\*\*\*\*

\*  
 \* PARTIE RAPPORT DE TEST \*  
 \*  
 \*  
 \*\*\*\*\*

\*\*\*\*\*

\*  
 \* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :  
 \*  
 \*  
 \*

\*\*\*\*\* Jeu de test JEU\_formatFichierDonnees : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_formatFichierDonnees : TEST\_OK.

\*\*\*\*\* CONCLUSION BILAN du test effectué le 17/09/08 : \*\*\*\*\*

\*  
 \* TEST\_OK,  
 \*  
 \* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.  
 \*  
 \*\*\*\*\*

\*\*\* Statut du SCENARIO DE TEST scnInterpolation4 : test à rejouer en non régression.

Le SCENARIO DE TEST scnInterpolation4 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*

(scnInterpolation4 ne pointe pas de jeux d'autres scénarios)

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnInterpolation5 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*

\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*  
 \* Copyright INRA, février 2006 \*  
 \*\*\*\*\*

\*\*\* \*\* Report de test \*\*\* \*\*

```
*****
*
*          PARTIE DESCRIPTION DE TEST
*
*****
```

```
*****
Description du test :
```

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : voir scnInterpolation1.

Les tests du scénario scnInterpolation5 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation5 concernent des vérifications dans une situation non nominale.

Informations sur la simulation effectuée en C++ :  
 Se reporter à scnInterpolation1 car la situation de test du scénario scnInterpolation5 est complètement identique à celle du scénario scnInterpolation1 SAUF que les fichiers des données à interpoler sont modifiés pour le test afin de se placer dans la situation incorrecte à vérifier. Cette différence (entre la situation de scnInterpolation5 et celle de scnInterpolation1) change le comportement du logiciel.

- Les fichiers des données à interpoler (voir specificiteConfigPortageMM2CPP.h) :  
 specificiteConfigPortageMM2CPP.h (configuration des fichiers non retouchée) : idem scnInterpolation1.

Le fichier lesEntreesCPP/donneesPluie.txt : est supprimé afin de provoquer la situation d'un fichier introuvable.

Le fichier lesEntreesCPP/donneesPluieDouble.txt : idem scnInterpolation1.

```
*****
Jeu de test JEU_formatFichierDonnees :
```

Définition/objectif :

Jeu de test vérifiant que les fichiers contenant les données à interpoler sont bien gérées : cas anormal d'un fichier introuvable. Il s'agit d'une erreur détectée dans la phase d'initialisation initialisation\_phase1 (présentée dans le fichier help\_specification dans "Initialisation du système").

Méthodes/opérations de vérification :

Préalable : Description des conséquences pour le logiciel d'une erreur détectée dans la phase d'initialisation initialisation\_phase1 : se reporter à "Cas d'erreurs, traces et codes d'erreurs" dans le fichier help\_specification.

Observation/analyse du fichier des traces lesSortiesTest/traceEcranTests.res :

- Vérifier que initialisation\_phase1 s'est mal déroulée et que l'erreur considérée (fichier introuvable) est relevée :
  - vérifier dans traceEcranTests.res l'affichage du message signalant l'erreur détectée " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME" avec le code erreur adéquat (message qui apparait en cas d'erreur survenue en initialisation\_phase1).
  - vérifier dans ce message que ce code erreur est : CR\_NOK\_FichierInvalide=14 (défini dans genericitePortageMM2CPP.h).
  - vérifier dans traceEcranTests.res l'affichage en plus (avant) du message de précision/explication concernant l'erreur rencontrée (le message n'apparait que si la clé de compilation CLE\_AVEC\_TRACE\_ECRAN\_DONNEES est activée) : " \*\*\*\*\* Fichier d'enti



XX

\*\*\* SCENARIO DE TEST scnInterpolation6 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

```
*****
* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++      *
* Copyright INRA, février 2006                                             *
*****
```

\*\*\* \*\* Rapport de test \*\*\* \*\*

```
*****
*
*          PARTIE DESCRIPTION DE TEST
*
*****
```

Description du test :

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : voir scnInterpolation1.

Les tests du scénario scnInterpolation6 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation6 concernent des vérifications dans une situation non nominale.

Informations sur la simulation effectuée en C++ :  
 Se reporter à scnInterpolation1 car la situation de test du scénario scnInterpolation6 est complètement identique à celle du scénario scnInterpolation1 SAUF que les fichiers des données à interpoler sont modifiés pour le test afin de se placer dans la situation incorrecte à vérifier. Cette différence (entre la situation de scnInterpolation6 et celle de scnInterpolation1) change les résultats de la simulation.

- Les fichiers des données à interpoler (voir specificiteConfigPortageMM2CPP.h) :

specificiteConfigPortageMM2CPP.h (configuration des fichiers non retouchée) : idem scnInterpolation1.

Le fichier lesEntreesCPP/donneesPluie.txt : idem scnInterpolation1.

Le fichier lesEntreesCPP/donneesPluieDouble.txt (contenu modifié) : Le fichier donneesPluieDouble.txt présente l'anomalie de contenir une ligne incomplète au milieu des autres. Le contenu du fichier donneesPluieDouble.txt est identique à celui du fichier du scnInterpolation1 SAUF que l'avant dernière donnée de la 4ème ligne est supprimée :

```
-> ligne incomplète (4ème) : 2250.0 777.7 80.0 120.0 444.4 666.6
                             (au lieu de : 2250.0 777.7 80.0 120.0 444.4 3250.0 666.6).
-> sur la ligne précédant la ligne incomplète (ie la 3ème ligne)
                             (3500.0 777.7 75.0 70.0 444.4 3375.0 666.6),
    I20=3500.0 pour instant t_I20=70.0 s, et
    I25=3375.0 pour instant t_I25=75.0 s.
```

\*\*\*\*\*  
 Jeu de test JEU\_formatFichierDonnees :

Définition/objectif :

Préalable : Description des conséquences pour le logiciel du fait qu'un fichier ( donneesPluieDouble.txt) contient une ligne incomplète (une donnée manquante à sa 4ème ligne) : Cette erreur a des conséquences en cours de déroulement de la simulation,

lors des traitements des entités interpolées (pour lesquels les données à interpoler sont lues dans des fichiers). Le logiciel ne repère pas la ligne exacte où il manque une donnée. A la place de la donnée manquante, il lit la donnée suivante, et continue à lire les données du fichier au fur et à mesure des besoins, avec un décalage d'une donnée (à partir de la donnée manquante et ainsi de suite jusqu'à la fin).

Jeu de test vérifiant comment sont gérés les fichiers contenant les données à interpoler : cas anormal d'un fichier (donneesPluieDouble.txt) qui contient au milieu des autres une ligne incomplète. Ce jeu contrôle la mauvaise lecture de données dans le fichier et le mauvais calcul d'entités interpolées. Plus précisément, il s'agit de vérifier :

- qu'il y a mauvaise lecture de données dans le fichier donneesPluieDouble.txt et mauvais calcul des entités interpolées et des autres entités qui sont en lien avec les données issues du fichier donneesPluieDouble.txt : I20, I25, Fpluie20, Fpluie25.

- que le mauvais format du fichier donneesPluieDouble.txt n'a pas de conséquence sur les entités interpolées et les autres entités qui sont sans lien avec les données issues du fichier donneesPluieDouble.txt : I1\_pluie, Fpluie1.

Méthodes/opérations de vérification : idem JEU\_calculEntiteInterpolee de scnInterpolation1 SAUF QUE :

L'identité de contenu des fichiers n'est à vérifier que jusqu'aux instants  $\leq t_{I20}$  (qui vaut 70.0) pour :

I\_attendu.res et I20.res  
F\_attendu.res et Fpluie20.res

L'identité de contenu des fichiers n'est à vérifier que jusqu'aux instants  $\leq t_{I25}$  (qui vaut 75.0) pour :

I\_attendu.res et I25.res  
F\_attendu.res et Fpluie25.res

\*\*\*\*\*

Jeu de test JEU\_ErreurNonDetectee :

Définition/objectif :

Préalable : voir ci-dessus le préalable du JEU\_formatFichierDonnees.

Jeu de test vérifiant comment sont gérés les fichiers contenant les données à interpoler : cas anormal d'un fichier (donneesPluieDouble.txt) qui contient au milieu des autres une ligne incomplète. Ce jeu observe que la simulation est déroulée sans détecter le mauvais format du fichier.

En effet, le logiciel tombe sur la fin de fichier alors qu'il tente de lire la dernière colonne de sa dernière ligne : alors qu'il lui a fallu lire une donnée dans le fichier, la donnée n'a pas été trouvée (a été mal lue), ce que le logiciel interprète comme fin de fichier, sachant que dans l'absolu cela peut correspondre soit à une erreur (cas comme ici où le fichier n'est pas au format attendu : nombre de colonnes, ligne incomplète...) soit à un cas normal de fin de fichier (où l'on sera amené à extrapoler à droite lors des calculs d'interpolation linéaire).

A RETENIR : Ce jeu fait ressortir le fait que le logiciel ne fait pas la distinction entre une fin de fichier normale et un problème de lecture fichier dû à mauvais format (une ligne incomplète...).

Méthodes/opérations de vérification :

Observation/analyse du fichier des traces lesSortiesTest/traceEcranTests.res :

Vérifier que le logiciel ne relève pas l'erreur considérée (mauvais format du fichier donneesPluieDouble.txt) ni aucune autre erreur : ABSENCE des messages " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVÉ A L'INITIALISATION DU SYSTEME" et " Fin de fonction simuler en erreur ...".

Vérifier que la simulation est bien lancée et se déroule correctement jusqu'à la fin :

- affichage des messages : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions d

e simulation demandées ..." et "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE ... les conditions de la simulation effectuée ...".

- affichage des messages : "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur" (signalant une fin correcte).

- vérifier dans lesSortiesTest que les fichiers résultats associés aux entités (.res) ont été générés.

\*\*\*\*\*

\*\*\*\*\*

\* \* \* \* \*

PARTIE RAPPORT DE TEST

\* \* \* \* \*

\*\*\*\*\*

\*\*\*\*\*

\* \* \* \* \*

\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\*\*\*\*\* Jeu de test JEU\_formatFichierDonnees : \*\*\*\*\*

Pour tous les fichiers comparés 2 à 2, la 1ère valeur diffère systématiquement, de la même manière que dans le jeu JEU\_calculEntiteInterpolee de scnInterpolation1. La différence sur la 1ère valeur ne peut pas être considérée comme une erreur, elle est due au décalage dans l'instant d'affichage initial, lié au démarrage (1er instant d'affichage = tDebutDeLaSimulation + deltat et non pas exactement tDebutDeLaSimulation). Pour plus de précisions, voir le rapport du jeu JEU\_calculEntiteInterpolee de scnInterpolation1.

A part cette 1ère valeur, les commandes "diff" ne relèvent aucune différence entre les fichiers comparés :

- dans leur totalité pour :
  - diff lesEntreesTest/t\_attendu.res lesSortiesTest/t.res
  - diff lesEntreesTest/I\_attendu.res lesSortiesTest/I1\_pluie.res
  - diff lesEntreesTest/F\_attendu.res lesSortiesTest/Fpluie1.res

- sur la plage des instants antérieurs à t\_I20=70.0 inclus, c'est-à-dire sur les 5 premières lignes des fichiers comparés (puisque t=70s en 5ème ligne et t=80s en 6ème ligne), pour :

- diff lesEntreesTest/I\_attendu.res lesSortiesTest/I20.res
- diff lesEntreesTest/F\_attendu.res lesSortiesTest/Fpluie20.res

- sur la plage des instants antérieurs à t\_I25=75.0 inclus, c'est-à-dire sur les 5 premières lignes des fichiers comparés (puisque t=70s en 5ème ligne et t=80s en 6ème ligne), pour :

- diff lesEntreesTest/I\_attendu.res lesSortiesTest/I25.res
- diff lesEntreesTest/F\_attendu.res lesSortiesTest/Fpluie25.res

Remarque : Dans le fichier d'entités à interpoler ../lesEntreesCPP/donneesPluieDouble.txt, le logiciel a identifié l'intervalle de temps des données à interpoler [ 55, 190 ] au lieu de [ 55, 115 ], ce qui reflète/révèle le décalage dans la lecture des données du fichier.

Conclusion du JEU\_formatFichierDonnees : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_ErreurNonDetectee : \*\*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), en particulier la simulation a été déroulée sans relever d'erreur en signalant une fin correcte, et la présence des fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_ErreurNonDetectee : TEST\_OK.

```

***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*
* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
*
*****

```

\*\*\* Statut du SCENARIO DE TEST scnInterpolation6 : test à rejouer en non régression.

Le SCENARIO DE TEST scnInterpolation6 a été créé pour vérifier la version 080917 du logiciel.

```

*****
(scenInterpolation6 ne pointe pas de jeux d'autres scénarios)
*****

```

XX

XX

\*\*\* SCENARIO DE TEST scnInterpolation7 : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

```

*****
* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++      *
* Copyright INRA, février 2006                                             *
*****

```

\*\*\* \*\* Report de test \*\*\* \*\*

```

*****
*
*          PARTIE DESCRIPTION DE TEST
*
*****

```

Description du test :

Ces tests font partie de la famille des tests portant sur le traitement d'interpolation, ie l'ensemble des scénarios { scnInterpolation\_i }.

Présentation de la FAMILLE DES SCENARIOS { scnInterpolation\_i } : voir scnInterpolation1.

Les tests du scénario scnInterpolation7 ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnInterpolation7 concernent des vérifications dans une situation non nominale.

Informations sur la simulation effectuée en C++ :  
Se reporter à scnInterpolation1 car la situation de test du scénario scnInterpolation7 est complètement identique à celle du scénario scnInterpolation1 SAUF que les fic



hiers des données à interpoler sont modifiés pour le test afin de se placer dans la situation incorrecte à vérifier. Cette différence (entre la situation de `scnInterpolation7` et celle de `scnInterpolation1`) change le comportement du logiciel.

- Les fichiers des données à interpoler (voir `specificiteConfigPortageMM2CPP.h`) :

`specificiteConfigPortageMM2CPP.h` (configuration des fichiers non retouchée) : idem `scnInterpolation1`.

Le fichier `lesEntreesCPP/donneesPluie.txt` : idem `scnInterpolation1`.

Le fichier `lesEntreesCPP/donneesPluieDouble.txt` (contenu modifié) : Le fichier `donneesPluieDouble.txt` présente l'anomalie de contenir sur une colonne utilisée 2 instants consécutifs identiques. Le contenu du fichier `donneesPluieDouble.txt` est identique à celui du fichier du `scnInterpolation1` SAUF que l'instant `t_I25` de la 4ème ligne est identique à celui de la 3ème ligne :

```
-> 3ème ligne (inchangée) : 3500.0 777.7 75.0 70.0 444.4 3375.0 666.6
-> 4ème ligne (changée)   : 2250.0 777.7 75.0 120.0 444.4 3250.0 666.6
      (au lieu de : 2250.0 777.7 80.0 120.0 444.4 3250.0 666.6).
```

```
I25=3375.0 pour instant t3_I25=75.0 s, et
I25=3250.0 pour instant t4_I25=75.0 s (au lieu de 80.0 s).
```

\*\*\*\*\*  
Jeu de test JEU\_valeursFichierDonnees :

Définition/objectif :

Jeu de test vérifiant que les fichiers contenant les données à interpoler sont bien gérés : cas anormal d'un fichier avec liste d'instants non strictement croissante. Il s'agit d'une erreur détectée dans la phase d'initialisation `initialisation_phasel` (présentée dans le fichier `help_specification` dans "Initialisation du système").

Méthodes/opérations de vérification :

Préalable : Description des conséquences pour le logiciel d'une erreur détectée dans la phase d'initialisation `initialisation_phasel` : se reporter à "Cas d'erreurs, traces et codes d'erreurs" dans le fichier `help_specification`.

Observation/analyse du fichier des traces `lesSortiesTest/traceEcranTests.res` :

Vérifier que `initialisation_phasel` s'est mal déroulée et que l'erreur considérée (le fait que le fichier ne contient qu'une seule ligne) est relevée :

- vérifier dans `traceEcranTests.res` l'affichage du message signalant l'erreur détectée " SIMULATION NON LANCEE POUR CAUSE DE PROBLEME RELEVE A L'INITIALISATION DU SYSTEME" avec le code erreur adéquat (message qui apparaît en cas d'erreur survenue en `initialisation_phasel`).

- vérifier dans ce message que ce code erreur est : `CR_NOK_FichierInvalide=14` (défini dans `genericitePortageMM2CPP.h`).

- vérifier dans `traceEcranTests.res` l'affichage en plus (avant) du message de précision/explication concernant l'erreur rencontrée (le message n'apparaît que si la clé de compilation `CLE_AVEC_TRACE_ECRAN_DONNEES` est activée) : " \*\*\*\*\* Fichier d'entités à interpoler : problème de liste d'instants dans le fichier non strictement croissante" pour le fichier `lesEntreesCPP/donneesPluieDouble.txt`.

- vérifier dans `traceEcranTests.res` l'affichage en plus (avant) du message de précision/explication concernant l'erreur rencontrée (le message n'apparaît que si la clé de compilation `CLE_AVEC_TRACE_ECRAN_DONNEES` est activée) : " \*\*\*\*\* Fichier d'entités à interpoler : non valide" pour le fichier `lesEntreesCPP/donneesPluieDouble.txt`.

Vérifier qu'après l'`initialisation_phasel`, le logiciel s'arrête sans que ne soit déroulée de simulation :

- ABSENCE d'affichage du message : " \*\*\*\*\* COMMANDE DE LA SIMULATION ... les conditions de simulation demandées ..." (message qui n'arrive à la suite de `initialisation_phasel` que si `initialisation_phasel` s'est bien passée).

- et à plus forte raison ABSENCE d'affichage des messages : "\*\*\*\*\* DEBUT DE LA SIMULATION EFFECTUEE" et "\*\*\*\*\* FIN DE LA SIMULATION" et "Fin de fonction simuler sans erreur".

- vérifier dans `lesSortiesTest` que les fichiers résultats associés aux entités (E

.res) n'ont pas été générés.

\*\*\*\*\*

\*\*\*\*\*
\*
\* PARTIE RAPPORT DE TEST
\*
\*\*\*\*\*

\*\*\*\*\*
\*
\* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
\*
\*\*\*\*\*

\*\*\*\* Jeu de test JEU\_valeursFichierDonnees : \*\*\*\*

Les affichages attendus dans le fichier lesSortiesTest/traceEcranTests.res ont bien été observés (messages présents, messages absents), parmi lesquels le message d'erreur attendu avec le bon code erreur. Comme attendu, il n'y a pas de fichiers résultats d'entités (.res) dans lesSortiesTest.

Conclusion du JEU\_valeursFichierDonnees : TEST\_OK.

\*\*\*\* CONCLUSION BILAN du test effectué le 17/09/08 : \*\*\*\*
\*
\* TEST\_OK,
\*
\* Tous les jeux de test ont été déroulés, aucune anomalie n'a été relevée.
\*
\*\*\*\*\*

\*\*\* Statut du SCENARIO DE TEST scnInterpolation7 : test à rejouer en non régression.

Le SCENARIO DE TEST scnInterpolation7 a été créé pour vérifier la version 080917 du logiciel.

\*\*\*\*\*

(scnInterpolation7 ne pointe pas de jeux d'autres scénarios)

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnPrecisionCalculs : fait partie des tests vérifiant le logiciel portageMM2CPP (leSourceCPP) \*\*\*

\*\*\*\*\*
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++
\* Copyright INRA, février 2006
\*\*\*\*\*

\*\*\* \*\* Rapport de test \*\*\* \*\*

\*\*\*\*\*  
 \*  
 \* PARTIE DESCRIPTION DE TEST \*  
 \*  
 \*\*\*\*\*

\*\*\*\*\*

Description du test :

Les vérifications du scénario scnPrecisionCalculs portent sur la précision des calculs effectués, plus particulièrement sur le calcul d'intégrale intervenant dans le traitement des entités de type entityAvecDerivee.

Les tests du scénario scnPrecisionCalculs ne s'appuient pas sur des résultats obtenus sous ModelMaker.

Les tests du scénario scnPrecisionCalculs concernent des vérifications dans une situation nominale.

Informations sur la simulation effectuée en C++ :  
 Se reporter à scnIntegrale car la situation de test du scénario scnPrecisionCalculs est complètement identique à celle du scénario scnIntegrale SAUF qu'ici on va "faire varier" le pas de temps d'horloge ddt. Plus exactement : Le scénario scnPrecisionCalculs n'est pas déroulé automatiquement, le fichier de configuration genericiteConfigPortageMM2CPP.h (réécrit pour les tests) est modifié en cours de déroulement du scénario.

Rappels :  
 La dérivée en fonction du temps de C\_constante ddt = 50 est une constante.  
 La dérivée en fonction du temps de C\_monte ddt = 200 + 30 t est une droite de pente ascendante.  
 La dérivée en fonction du temps de C\_descend ddt = 6200 - 30 t est une droite de pente descendante.  
 La dérivée en fonction du temps de C\_sinusoidale ddt = sin( t x PI/20 - 2PI) est une sinusoïde comportant 3 périodes entre t=40 s et t=160 s ; nulle à t = 40, 60, 80, 100, 120, 140 et 160 s ; max à t = 50, 90, 130 s ; min à t = 70, 110, 150 s.  
 Pour plus de renseignements, voir "lesEntreesTest/readme".

\*\*\*\*\*

Jeu de test JEU\_precisionCalculs :

Définition/objectif :

Jeu de test évaluant (donnant des éléments de mesure de) la précision du calcul d'intégration.

Méthodes/opérations de vérification :

Ce jeu se déroule en 3 étapes, il s'appuie sur le jeu JEU\_calculValueNumerique du scénario scnIntegrale. Plus précisément :

- 1) Simulation sous ddt = 0,001 s :
  - Se placer dans les conditions du scénario scnIntegrale (selon lesquelles dans le fichier de configuration genericiteConfigPortageMM2CPP.h réécrit pour les tests : ddt=0.001 s).
  - Jouer la partie "2) Vérification fine" du jeu de test JEU\_calculValueNumerique du scénario scnIntegrale.
  - Noter les écarts trouvés, afin d'en garder la trace.
- 2) Simulation sous ddt = 0,0001 s :
  - Se placer dans les conditions du scénario scnIntegrale.
  - Dans le fichier de configuration genericiteConfigPortageMM2CPP.h réécrit pour les tests, mettre le pas de temps d'horloge : ddt=0.0001 s.
  - Jouer la partie "2) Vérification fine" du jeu de test JEU\_calculValueNumerique

du scénario scnIntegrale.

- Noter les écarts trouvés, afin d'en garder la trace.

- Dans le fichier de configuration genericiteConfigPortageMM2CPP.h réécrit pour les tests, remettre le pas de temps d'horloge : deltat=0.001 s.

3) Simulation sous deltat = 0,00001 s :

- Se placer dans les conditions du scénario scnIntegrale.

- Dans le fichier de configuration genericiteConfigPortageMM2CPP.h réécrit pour les tests, mettre le pas de temps d'horloge : deltat=0.00001 s.

- Jouer la partie "2) Vérification fine" du jeu de test JEU\_calculValueNumerique du scénario scnIntegrale.

- Noter les écarts trouvés, afin d'en garder la trace.

- Dans le fichier de configuration genericiteConfigPortageMM2CPP.h réécrit pour les tests, remettre le pas de temps d'horloge : deltat=0.001 s.

\*\*\*\*\*

\*\*\*\*\*

```
*
*
*          PARTIE RAPPORT DE TEST          *
*
*
*****
```

\*\*\*\*\*

```
*
*
* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 17/09/08 :
*
*
*
*****
```

\*\*\*\*\* Jeu de test JEU\_precisionCalculs : \*\*\*\*\*

Rappel des conditions du scénario scnIntegrale : Simulation de l'instant tDebutDeLaSimulation=40 s à l'instant tMaxDeLaSimulation=160 s, avec un pas de simulation pasSimulation=1 s, et un pas de temps d'horloge deltat=0.001 s et PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN=1000.

Lors de la "2) Vérification fine" (où seuls sont pris en compte les couples (valeur de C\_xxx, valeur de C\_xxx\_attendu) relatifs aux instants tels que t et t\_attendu coïncident exactement), on calcule les écarts max (absolus) suivants entre valeurs issues des calculs/traitements (fichiers lesSortiesTest/C\_xxx.res) et valeurs attendues/précalculées pour test (fichiers lesEntreesTest/C\_xxx\_attendu.res). Ces écarts reflètent fidèlement la précision des calculs (dans des cas précis : droite, sinusoïde).

1) Simulation sous deltat = 0,001 s (tient compte des instants entre t=41 s et t = 160 s inclus) :

Pour C\_constante :

MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000000

Pour C\_monte :

MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 1,812500

Pour C\_descend :

MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 1,812500

Pour C\_sinusoidale :

MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000502

Analyse : Les résultats obtenus ne sont pas très satisfaisants pour C\_monte et C\_descend (ddt droites de pente +/-30). Les résultats obtenus sont satisfaisants pour C\_constante et C\_sinusoidale.

Remarque : Sachant que  $\text{deltat}=0.001$  s et  $\text{PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN}=1000$  (cf fichier `genericiteConfigPortageMM2CPP.h`), le pas de simulation  $\text{pasSimulation}$  1 s (= 0.001 s x 1000) est minimal, c'est à dire qu'on est dans la situation avec la précision de calcul la plus faible. Les résultats montrent que  $\text{PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN}=1000$  n'est pas assez exigeant dans le contexte considéré.

2) Simulation sous  $\text{deltat} = 0,0001$  s (tient compte des instants entre  $t=70$  s et  $t = 160$  s inclus) :

```

    Pour C_constante :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000000
    Pour C_monte :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,187500
    Pour C_descend :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,187500
    Pour C_sinusoidale :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000086
    
```

Analyse : Les résultats obtenus deviennent satisfaisants pour C\_monte et C\_descend (ddt droites de pente +/-30). Ils le sont toujours pour C\_constante et C\_sinusoidale.

3) Simulation sous  $\text{deltat} = 0,00001$  s (tient compte des instants entre  $t=68$  s et  $t = 160$  s inclus) :

```

    Pour C_constante :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000000
    Pour C_monte :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,031250
    Pour C_descend :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,031250
    Pour C_sinusoidale :
    MAX( | ECART ( donnée calculée provenant de lesSortiesTest, donnée attendue provenant de lesEntreesTest) | ) = 0,000050
    
```

Analyse : Les résultats obtenus sont améliorés.

Conclusion du JEU\_precisionCalculs :

Les résultats incitent à la recommandation suivante : pour une simulation de l'instant  $t\text{DebutDeLaSimulation}=40$  s à l'instant  $t\text{MaxDeLaSimulation}=160$  s, avec un pas de simulation  $\text{pasSimulation}=1$  s, où sont traités les compartiments C\_constante, C\_monte, C\_descend et C\_sinusoidale : mettre comme pas de temps d'horloge  $\text{deltat}=0.0001$  s (ceci reviendrait à augmenter  $\text{PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN}=10000$  au lieu de 1000), ne pas laisser  $\text{deltat}=0.001$  s qui est insuffisant.

```

***** CONCLUSION BILAN du test effectué le 17/09/08 : *****
*
* TEST_OK,
*
* Tous les jeux de test ont été déroulés, aucune anomalie/aberration n'a été relevé
e.
*
*****
    
```

\*\*\* Statut du SCENARIO DE TEST `scnPrecisionCalculs` : test à rejouer en non régression.

Le SCENARIO DE TEST `scnPrecisionCalculs` a été créé pour vérifier la version 080917 d

u logiciel.

Ce scénario permet d'analyser les performances de calcul en terme de précision. Il peut servir à déceler une insuffisance de précision, à ajuster/adapter la constante PAS\_SIM\_SUR\_PAS\_TEMPS\_MIN (valeur minimale du rapport pasDeLaSimulation/deltat) au cas du modèle traité.

Attention : Le scénario scnPrecisioncalculs comprend des interventions manuelles "en cours de simulation". Il n'est pas "déroulé automatiquement" dans le sens où l'opérateur testeur doit modifier le fichier de configuration genericiteConfigPortageMM2CPP.h (réécrit pour les tests) en cours de déroulement du scénario.

\*\*\*\*\*

scnPrecisionCalculs pointe des jeux de :  
- scnIntegrale

\*\*\*\*\*

XX

XX

\*\*\* SCENARIO DE TEST scnTestV071025 : fait partie des tests vérifiant que le logiciel portageMM2CPP écrit en C++ (leSourceCPP) est conforme au modèle sous sa forme ModelMaker (modele2.mod) \*\*\*

\*\*\*\*\*  
\* Logiciel portageMM2CPP de portage de code ModelMaker en langage C++ \*  
\* Copyright INRA, février 2006 \*  
\*\*\*\*\*

\*\*\* \*\* Report de test \*\*\* \*\*

\*\*\*\*\*  
\*  
\* PARTIE DESCRIPTION DE TEST \*  
\*  
\*\*\*\*\*

\*\*\*\*\*  
Description du test :

Les tests de la version precedente 060203 n'ont pas ete rejoues.

Les changements et la non regression du logiciel dans sa version 071025 ont ete testés en vérifiant que les fichiers de lesSorties obtenus a l'execution etaient identiques a ceux de la version precedente 060203 : verification des resultats par diff \*.res entre nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/enCours\_logiciel\_portageMM2CPP/lesSortiesCPP et les resultats obtenus sous cygwin avec la version precedente nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_060203\_portageMM2CPP/logiciel\_portageMM2CPP/lesSortiesCPP.

\*\*\*\*\*

\*\*\*\*\*  
\*  
\* PARTIE RAPPORT DE TEST \*  
\*\*\*\*\*



Les tests de la version 060203 n'ont pas ete rejoues.

Les changements et la non regression du logiciel dans sa version 071026 ont ete test es en comparant les fichiers \*.res de lesSorties obtenus a l'execution (qui se trouv ent dans nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/enCours\_logiciel\_portageMM2C PP/lesSortiesCPP) avec ceux de la version precedente 071025 (qui se trouvent dans nr ousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_071025\_portageMM2CPP/logiciel\_ portageMM2CPP/lesSortiesCPP).

+ un test de capacite.

\*\*\*\*\*

\*\*\*\*\*

Jeu de test JEU\_m\_bool :

Définition/objectif :

Jeu de test vérifiant que la modification m\_bool n'a rien change aux comportement s/resultats du logiciel.

Méthodes/opérations de vérification :

La verification est effectuee alors que les modifications m\_bool et m\_type ont et e effectuees, et que le code est dans l'etat : "typedef float typeTemps;" et "typed e float typeValeur;".

Verification des resultats par diff \*.res entre nrousse@badet : ~/myworkspace/WI\_ portageMM2CPP/enCours\_logiciel\_portageMM2CPP/lesSortiesCPP et les resultats obtenus avec la version precedente nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_0 71025\_portageMM2CPP/logiciel\_portageMM2CPP/lesSortiesCPP.

\*\*\*\*\*

Jeu de test JEU\_m\_type\_double :

Définition/objectif :

Jeu de test vérifiant la modification m\_type et que la passage au type double n'a pas degrade les resultats du logiciel.

Méthodes/opérations de vérification :

La verification est effectuee alors que les modifications m\_bool et m\_type ont et e effectuees, et que le code est dans l'etat : "typedef double typeTemps;" et "typed ef double typeValeur;".

Verification des resultats par diff \*.res entre nrousse@badet : ~/myworkspace/WI\_ portageMM2CPP/enCours\_logiciel\_portageMM2CPP/lesSortiesCPP et les resultats obtenus avec la version precedente nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_0 71025\_portageMM2CPP/logiciel\_portageMM2CPP/lesSortiesCPP.

\*\*\*\*\*

Jeu de test JEU\_m\_type\_double\_graphique :

Définition/objectif :

Jeu de test vérifiant la modification m\_type et que la passage au type double n'a pas degrade les resultats du logiciel.

Méthodes/opérations de vérification :

La verification est effectuee alors que les modifications m\_bool et m\_type ont et e effectuees, et que le code est dans l'etat : "typedef double typeTemps;" et "typed ef double typeValeur;".

Comparaison visuelle de graphiques. Les données à vérifier sont :



- Données \*.res de nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/enCours\_logiciel\_portageMM2CPP/lesSortiesCPP  
 - Données \*.res de nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_071025\_portageMM2CPP/logiciel\_portageMM2CPP/lesSortiesCPP.

Note 1. : ce graphique peut être tracé à l'écran à l'aide de l'outil "affgraphs.py". "affgraphs.py" est un logiciel d'affichage de graphiques (il propose et gère une interface permettant de configurer/paramétrer des représentations graphiques, puis affiche à l'écran les tracés demandés). Il en a été fait une copie "copie\_affgraphs.py" dans "lesSorties" afin de l'y utiliser ( commande : "python affgraphs.py" ) ; pour plus d'informations (guide d'utilisation, spécification ...), consulter entêtes et commentaires du fichier "affgraphs.py".

\*\*\*\*\*  
 Jeu de test JEU\_m\_type\_long\_double :

Définition/objectif :

Jeu de test vérifiant la modification m\_type et que la passage au type long double n'a pas dégradé les résultats du logiciel.

Méthodes/opérations de vérification :

La vérification est effectuée alors que les modifications m\_bool et m\_type ont été effectuées, et que le code est dans l'état : "typedef long double typeTemps;" et "typedef long double typeValeur;".

Vérification des résultats par diff \*.res entre nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/enCours\_logiciel\_portageMM2CPP/lesSortiesCPP et les résultats obtenus avec la version précédente nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_071025\_portageMM2CPP/logiciel\_portageMM2CPP/lesSortiesCPP.

\*\*\*\*\*  
 Jeu de test JEU\_m\_type\_long\_double\_graphique :

Définition/objectif :

Jeu de test vérifiant la modification m\_type et que la passage au type long double n'a pas dégradé les résultats du logiciel.

Méthodes/opérations de vérification :

La vérification est effectuée alors que les modifications m\_bool et m\_type ont été effectuées, et que le code est dans l'état : "typedef long double typeTemps;" et "typedef long double typeValeur;".

Comparaison visuelle de graphiques. Les données à vérifier sont :

- Données \*.res de nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/enCours\_logiciel\_portageMM2CPP/lesSortiesCPP  
 - Données \*.res de nrousse@badet : ~/myworkspace/WI\_portageMM2CPP/photoTmp\_071025\_portageMM2CPP/logiciel\_portageMM2CPP/lesSortiesCPP.

voir Note 1.

\*\*\*\*\*  
 Jeu de test JEU\_anomalie1 :

Définition/objectif :

Jeu de test vérifiant que l'anomalie Anomalie.1 est corrigée.

Méthodes/opérations de vérification :

La vérification est effectuée alors que la modification m\_type a été effectuée.

Pour les besoins du test, le code source est retouché pour se placer dans les conditions où l'anomalie Anomalie.1 se produit :

-----

```

| #define DELTA_T_DEFAUT 1.0 // pas de temps par défaut : 1.0 seconde
| Appel simulation :
| /* 1 jour = 24 x 60 x 60 s = 86400 s */
| S.simuler( 200, 86400.0 ); // 200 itérations (ie 200 jours), pas de simula
tion 86400 s (ie 1 jour)
-----

```

Le test enchaîne 3 essais qui nécessitent de retoucher le code entre 2 :

1er essai, avec "typedef long double typeTemps;" et "typedef long double typeValeur;" : vérifier que le logiciel se déroule correctement jusqu'au dernier jour (200).

2nd essai, avec "typedef float typeTemps;" et "typedef float typeValeur;" : vérifier que le logiciel bloque au jour 194 (reproduction de l'anomalie) ie ne se déroule pas correctement jusqu'au dernier jour (200).

3eme essai, avec "typedef double typeTemps;" et "typedef double typeValeur;" : vérifier que le logiciel se déroule correctement jusqu'au dernier jour (200).

\*\*\*\*\*  
 Jeu de test JEU\_capacite :

Définition/objectif :

Jeu de test qui a été fait pour analyser les circonstances dans lesquelles se produisait l'anomalie Anomalie.1 et comment la corriger.

Méthodes/opérations de vérification :

Le test est effectuée alors que les modifications m\_bool et m\_type n'ont pas encore été effectuées. Le code est dans l'état :

```

-----
| class horlogeLocale { public : double t; double tInitial; double deltat; ...
}
| class entityInstantanee { public : float value; float initialValue; ... }
| class entityAvecDerivee : public entityInstantanee { public : float ddt; ...
}
-----
| #define DELTA_T_DEFAUT 1.0 // pas de temps par défaut : 1.0 seconde
| Appel simulation :
| /* 1 jour = 24 x 60 x 60 s = 86400 s */
| S.simuler( 73000, 86400.0 ); // 73000 itérations (ie 200 ans), pas de simu
lation 86400 s (ie 1 jour)
-----

```

Vérifier que la simulation se déroule correctement "suffisamment longtemps", c'est à dire que le logiciel ne bloque pas (production de l'anomalie) en cours d'exécution.

\*\*\*\*\*

```

*****
*
*                               PARTIE RAPPORT DE TEST
*
*****

```

\*\*\*\*\*

```

*
* RAPPORT/CONCLUSIONS DU TEST EFFECTUE LE 26/10/07 :
*
*
*

```

\*\*\*\*\* Jeu de test JEU\_m\_bool : \*\*\*\*\*

Les diff n'indiquent aucune différence.

Conclusion du JEU\_m\_bool : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_m\_type\_double : \*\*\*\*\*

Les diff indiquent des differences quasi-partout. Toutefois, a premiere vue, ces dif ferences semblent correspondre a la difference de precision entre des calculs float et des calculs double.

Conclusion du JEU\_m\_type\_double : TEST\_NOK qui sera considere comme acceptable ou no n en fonction du resultat du jeu de test JEU\_m\_type\_double\_graphique.

\*\*\*\*\* Jeu de test JEU\_m\_type\_double\_graphique : \*\*\*\*\*

Visuellemnt, les traces coincident tres exactement.

Conclusion du JEU\_m\_type\_double\_graphique : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_m\_type\_long\_double : \*\*\*\*\*  
\*

Les diff indiquent des differences quasi-partout. Toutefois, a premiere vue, ces dif ferences semblent correspondre a la difference de precision entre des calculs float et des calculs long double.

Conclusion du JEU\_m\_type\_long\_double : TEST\_NOK qui sera considere comme acceptable ou non en fonction du resultat du jeu de test JEU\_m\_type\_long\_double\_graphique.

\*\*\*\*\* Jeu de test JEU\_m\_type\_long\_double\_graphique : \*\*\*\*\*

Visuellemnt, les traces coincident tres exactement.

Conclusion du JEU\_m\_type\_long\_double\_graphique : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_anomalie1 : \*\*\*\*\*

1er essai : l'execution va bien jusqu'au bout, le logiciel se deroule correctemen t jusqu'au dernier jour (200).

2nd essai : l'execution bloque au jour 194, le logiciel ne se deroule pas correct ement jusqu'au dernier jour (200).

3eme essai : l'execution va bien jusqu'au bout, le logiciel se deroule correcteme nt jusqu'au dernier jour (200).

Conclusion du JEU\_anomalie1 : TEST\_OK.

\*\*\*\*\* Jeu de test JEU\_capacite : \*\*\*\*\*

L'execution s'est deroulee correctement (au moins) jusqu'au jour 61000 (ie sur plus de 150 ans), ce qui a pris/dure environ 7 heures (parce qu'elle est longue, la simul ation est arretee avant la fin mais il ne s'est pas produit d'erreur d'execution).

Conclusion du JEU\_capacite : TEST\_OK.

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\* CONCLUSION BILAN du test effectue le 26/10/07 : \*\*\*\*\*

- \* TEST\_OK,
- \* Tous les jeux de test sont TEST\_OK sauf les jeux de test JEU\_m\_type\_double et JEU\_m\_type\_long\_double, qui sont quand meme acceptes car les jeux de tests JEU\_m\_type\_double\_graphique et JEU\_m\_type\_long\_double\_graphique sont TEST\_OK.
- \* Conclusion : modification m\_bool effectuee et validee
- \* => amelioration Amelioration.2 effectuee (close).
- \* Conclusion : modification m\_type effectuee et verifiee

