

*** 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                                           *
*****
```

*** ** Raport 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 deltata=0.001 s (la configuration dans le fichier genericiteConfigPortageMM2CPP.h n'est pas retouchée pour le test). Remarque : par rapport à deltata 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=40s) est 7.0 et
- sa fonction dérivée ddt est : ddt = 50

Compartiment C_monte :

- sa valeur "initiale" (ie à t=40s) est 0.0 et
- sa fonction dérivée ddt est : 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=40s) est 0.0 et
- sa fonction dérivée ddt est : 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=40s) est 0.0 et
- sa fonction dérivée ddt est : ddt = sin(t x PI/20 - 2PI)

- 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 s valeurs d'entités de type entityAvecDerivee : value des 4 compartiments (compartme nt) 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 visuelle de graphiques (voir "Note 1."). En plaçant en abscisses de tous les graphiques les données lesEntreesTest/t_attendu.res, les données à plac er 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 r ésultats 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.

Note 1. : les graphiques peuvent être tracés à l'écran à l'aide de l'outil "affgraph s.py". "affgraphs.py" est un logiciel d'affichage de graphiques (il propose et gère une interface permettant de configurer/paramétrer des représentations graphiques, pu is affiche à l'écran les tracés demandés). Il en a été fait une copie "copie_affgrap hs.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 (compartement) C_constante, C_monte, C_descend, C_sinusoida le 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_mon te, 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 (va leur 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 pouss ée :

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 Δt 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 $t_{attendu} < t$ (et même $t = t_{attendu} + \Delta t$).

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) | ) = 0,000000
```

nant de lesEntreesTest) |) = 1,812500
 Pour C_sinusoidale :
 MAX(| ECART (donnée calculée provenant de lesSortiesTest, donnée attendue prove
 nant de lesEntreesTest) |) = 0,000502

On considère que le niveau de précision des calculs justifie ces écarts (sans reg
 arder dans le détail l'influence des décalages entre t et t_attendu).

2) Vérification fine (seuls sont pris en compte les couples (valeur de C_xxx, val
 eur de C_xxx_attendu) relatifs aux instants tels que t et t_attendu coïncident exact
 ement) :

On calcule comme écarts max (absolus) entre valeurs issues des calculs/traitement
 s (fichiers lesSortiesTest/C_xxx.res) et valeurs attendues/précalculées pour test (f
 ichiers lesEntreesTest/C_xxx_attendu.res) :

Pour C_constante :
 MAX(| ECART (donnée calculée provenant de lesSortiesTest, donnée attendue prove
 nant de lesEntreesTest) |) = 0,000000
 Pour C_monte :
 MAX(| ECART (donnée calculée provenant de lesSortiesTest, donnée attendue prove
 nant de lesEntreesTest) |) = 1,812500
 Pour C_descend :
 MAX(| ECART (donnée calculée provenant de lesSortiesTest, donnée attendue prove
 nant de lesEntreesTest) |) = 1,812500
 Pour C_sinusoidale :
 MAX(| ECART (donnée calculée provenant de lesSortiesTest, donnée attendue prove
 nant 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 : d
 roite, 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.
 *
