

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

```
/*
 * Licence :
```

Le logiciel portageMM2CPP est une réécriture en langage C++ du modèle écrit dans le fichier ModelMaker "modele2.mod" (situé dans le répertoire "leSourceModelMaker"). Voir informations dans ../laDocumentation/help.

Ce logiciel est régi par la licence CeCILL soumise au droit français et respectant les principes de diffusion des logiciels libres. Vous pouvez utiliser, modifier et/ou redistribuer ce programme sous les conditions de la licence CeCILL telle que diffusée par le CEA, le CNRS et l'INRIA sur le site "<http://www.cecill.info>".

En contrepartie de l'accessibilité au code source et des droits de copie, de modification et de redistribution accordés par cette licence, il n'est offert aux utilisateurs qu'une garantie limitée. Pour les mêmes raisons, seule une responsabilité restreinte pèse sur l'auteur du programme, le titulaire des droits patrimoniaux et les concédants successifs.

A cet égard l'attention de l'utilisateur est attirée sur les risques associés au chargement, à l'utilisation, à la modification et/ou au développement et à la reproduction du logiciel par l'utilisateur étant donné sa spécificité de logiciel libre, qui peut le rendre complexe à manipuler et qui le réserve donc à des développeurs et des professionnels avertis possédant des connaissances informatiques approfondies. Les utilisateurs sont donc invités à charger et tester l'adéquation du logiciel à leurs besoins dans des conditions permettant d'assurer la sécurité de leurs systèmes et ou de leurs données et, plus généralement, à l'utiliser et l'exploiter dans les mêmes conditions de sécurité.

Le fait que vous puissiez accéder à cet en-tête signifie que vous avez pris connaissance de la licence CeCILL, et que vous en avez accepté les termes.

Les fichiers Licence_CeCILL_V2-fr.txt et Licence_CeCILL_V2-en.txt du répertoire ../LICENCE fournissent le texte de la licence, dans sa version 2, en langue française et en langue anglaise. (ces 2 fichiers sont aussi directement dans le répertoire courant)

```
*/
```

```
/*
*****
*
* Fichier      : specificitePortageMM2CPP.h
*
* Auteur(s)   : Nathalie Rousse, Nathalie.Rousse@toulouse.inra.fr
*               de l'INRA - Institut National de la Recherche Agronomique -
*               (département MIA, UMR AGIR, http://www.modelia.org).
*
* Description :
*
* Ce fichier fait partie du code source C++ du logiciel portageMM2CPP.
*
* Il en contient la partie complètement spécifique du modèle traité
* (déclarations des classes). Voir ../laDocumentation/help, en particulier
* "Présentation générale" et "Spécification".
*
*****
* Historique :
*
* 03/02/06, Nathalie Rousse : création du fichier.
*
* 02/01/08, Nathalie Rousse : modification m_linearInterpolation (ajout prise en compte des entités ModelMaker qui son
t calculées par interpolation linéaire à partir de données lues dans un fichier de données).
* 02/01/08, Nathalie Rousse : modification m_integrale.
*
*****/

/* modeDemploi
* Instruction s'adressant à quelqu'un qui est en train de construire son propre logiciel C++ relativement à son propre
modèle ModelMaker : dans tout le code de ce fichier, consulter et suivre les consignes/indications marquées du label "
modeDemploi". Voir aussi "Mode d'emploi" dans le fichier "help".
*/

/* modeDemploi
*
* Le code de ce fichier (et de son ".cpp") est D'OFFICE à réécrire.
*
* Par ailleurs, ce code est susceptible de bouger au titre d'évolutions : dans tout le code de ce fichier (et son ".cp
p"), voir les commentaires marqués du label "modeDemploiEvolutions".
*/

/* modeDemploiEvolutions
* Quelqu'un peut être conduit à reprendre les fonctions qui restituent les résultats sortis des simulations : par exem
ple s'il a besoin de sortir des résultats à d'autres formats ... ( cf "afficherEntetesEntites" et "afficherEntites de
la classe systemParticulier)
*/
```

```
#ifndef _SPECIFICITE_H_INCLUS_
#define _SPECIFICITE_H_INCLUS_

#include "genericitePortageMM2CPP.h"

#include "specificiteConfigPortageMM2CPP.h"

/* Constantes associées (voir fichier specificiteConfigPortageMM2CPP.h) :
 * nomFichier_DONNEE_I1_pluie, nbColonnes_DONNEE_I1_pluie,
 * colInstant_DONNEE_I1_pluie, colValeur_DONNEE_I1_pluie,
 * ..._I..., ..._I... */

/*****
 *
 * classe spécifiant le système particulier
 *
 *****/

/* modeDemploi
 * Cette classe est à réécrire : données déclarées et contenu des méthodes. Pour cette réécriture, s'aider des explications et illustrations (commentaires sur fond jaune) du "Diagramme des classes spécifiques" des représentations UML.
 */

/* m_linearInterpolation : des entités controlled sont ajoutées dans ce système */
class systemParticulier : public genericSystem {

    public :
        /*
         * Déclaration des éléments ModelMaker
         * composant le système particulier
         */

// modeDemploiDebut

        /* constantes */
        constant C3max;

        /* paramètres */
        parameter P1, P2, P3, P4, P5;

        /* controlled (in et interpolated) */
```

```
    controlled I1_pluie;
    controlled I20;
    controlled I25;

    /* variables */
    variable V1;

    /* flux */
    flow Fpluie1, Fpluie20, Fpluie25;
    flow F1, F2, F3, F4;

    /* compartiments */
    compartment C_constante, C_monte, C_descend, C_sinusoidale;
    compartment C1_source, C2_reservoir, C3_reservoir, C4_puits, C5_debordement;

/* modeDemploi
    *
    * remplacer le code situé entre ici et "modeDemploiDebut" ci-dessus par le code adapté à son système p
    articulier : se conformer au MAIN extrait du fichier ModelMaker (les déclarations), s'aider des commentaires sur fond j
    aune des représentations UML. */

    public :
        // Construction/initialisation avec horloge locale par défaut
        systemParticulier( void );

        /* Construction/initialisation
        * avec configuration de l'horloge locale :
        * instant initial et pas de temps (plus petit increment) */
        systemParticulier( typeTemps it, typeTemps deltatValue );

    private :

        /*
        * Initialisation des entités du système
        * Des résultats sont retournés (bien que return et paramètres : void).
        * Les résultats de initialiserLesEntites sont retournés dans crInit.
        */
        void initialiserLesEntites( void ); // appelé par les constructeurs

        // Evolution en fonction de l'incrément de temps
        int actualiserIncrement( void );

        /*
        * Affichages, restitution des résultats sortis des simulations
        */
```

```

        * Description de la forme actuelle de restitution des
        * résultats : il est fait des affichages à l'écran et dans
        * des fichiers résultats. Voir détails dans le ".cpp"
        */
        int afficherEntetesEntites(void);
        int afficherEntites(void);
};

#endif /* _SPECIFICITE_H_INCLUS_ */

/* modeDemploiEvolutions
 * Pour écrire la classe systemParticulier, il est pratique de s'appuyer sur le texte du MAIN de ModelMaker (extrait du
 * fichier modele2.mod). L'écriture de la classe systemParticulier est actuellement complètement manuelle, mais il y aura
 * it moyen d'automatiser - au moins partiellement - la tâche, par génération automatique de lignes C++ à partir du texte
 * du MAIN de ModelMaker.
 */

/* modeDemploi
 * Remplacer le commentaire ci-dessous par celui qui correspond à son propre modèle (utile pour effectuer l'adaptation
 * de la classe systemParticulier */

/*****
 *
 * PHOTO DU MAIN de ModelMaker (extrait du fichier modele2.mod)
 *
 * (il manque les paramètres : p1, p2, p3, p4, p5)
 */

Main
compartment: C1_source Unconditional
dC1_source/dt = -F1
Initial Value = 1
compartment: C2_reservoir Unconditional
dC2_reservoir/dt = +F1-F2
Initial Value = 0.0
compartment: C3_reservoir Unconditional
dC3_reservoir/dt = +F2-F3-F4
Initial Value = 0.0
define value: C3max Unconditional Global
C3max = 0.2
compartment: C4_puits Unconditional
dC4_puits/dt = +F3
Initial Value = 0.0

```

```

compartment: C5_debordement Unconditional
dC5_debordement/dt = +F4
Initial Value = 0.0
flow: F1 Unconditional
Flow from C1_source to C2_reservoir
F1 = (p1*t/(t+p4)) * C1_source
flow: F2 Unconditional
Flow from C2_reservoir to C3_reservoir
F2 = V1 * C2_reservoir
flow: F3 Unconditional
Flow from C3_reservoir to C4_puits
F3 = p3 * C3_reservoir
flow: F4 Conditional
Flow from C3_reservoir to C5_debordement
F4 =
  p5 * ( C3_reservoir - C3max ) for C3_reservoir >= C3max
  0 by default
variable: V1 Unconditional
V1 = p2 * ( 1 + sin(t/5) )

```

// m_linearInterpolation : en plus des entités du modèle ModelMaker modele2.mod (ie définies dans le Main ci-dessus), des entités controlled ont été ajoutées au système/modèle (controlled I1_pluie, I20, I25) et vis à vis de ces entités controlled, des entités flux (flow Fpluie1, Fpluie20, Fpluie25). I1_pluie, I20, I25 correspondent à la même donnée réelle mais sont lues dans 3 colonnes distinctes, ne proviennent pas toutes du même fichier, ne reposent pas toutes sur une seule même liste d'instants. flow Fpluie1, Fpluie20 et Fpluie25 sont calculés à partir respectivement de I1_pluie, I20, I25 selon la même "formule".

// m_integrale : en plus des entités du modèle ModelMaker modele2.mod (ie définies dans le Main ci-dessus), des compartiments ont été ajoutés au système/modèle (compartiment C_constante, C_monte, C_descend, C_sinusoidale pour mettre en oeuvre le calcul d'intégrale. Ces ajouts ont permis de repérer des modifications nécessaires, qui ont été effectuées.

*

*

*****/

/******

*

*

* Fin du fichier : specificitePortageMM2CPP.h

*

*

*

*****/