



Direction ÉNERGIE ENVIRONNEMENT

Division Développements Informatique et Méthodes

Numériques

Guide de programmation des extensions dynamiques RT2012

La reproduction de ce rapport n'est autorisée que sous sa forme intégrale, sauf clauses spécifiques explicitées dans la convention liant le CSTB et le donneur d'ordre.

Toute reproduction, même partielle, devra mentionner le CSTB et le ou les auteurs.

Il comporte 89 pages

NOVEMBRE 2016

DEE/DIMN-2013.009RE

CENTRE SCIENTIFIQUE ET TECHNIQUE DU BATIMENT

SIÈGE SOCIAL > 84 AVENUE JEAN JAURÈS | CHAMPS-SUR-MARNE | 77447 MARNE-LA-VALLÉE CEDEX 2

TÉL. (33) 01 64 68 82 82 | FAX. (33) 01 60 05 70 37 | SIRET 775 688 229 000 27 | www.cstb.fr

ÉTABLISSEMENT PUBLIC À CARACTÈRE INDUSTRIEL ET COMMERCIAL | RCS MEAUX 775 688 229 | TVA FR 70 775 688 229

MARNE-LA-VALLÉE | PARIS | GRENOBLE | NANTES | SOPHIA-ANTIPOLIS

DIFFUSION RESTREINTE

**PAGE
LAISSEE
INTENTIONNELLEMENT
VIDE**

SOMMAIRE

1. SPECIFICATIONS GENERALES	5
1.1 Introduction	5
1.2 A propos des extensions dynamiques	5
1.3 Structuration de l'extension.....	7
2. CREER UNE EXTENSION	9
2.1 Remarque liminaire	9
2.2 Conventions	9
2.3 Classe de paramètres	9
2.3.1 Implémenter l'interface du jeu de paramètres utilisateurs	9
2.3.2 Ajouter la lecture du jeu de paramètres depuis un fichier XML	12
2.3.3 exposer une extension du jeu de paramètres d'entrée	13
2.3.4 Ajouter la descriptions des paramètres	14
2.3.5 Génération du XSD.....	17
2.4 L'algorithme	19
2.4.1 A propos du code de calcul horaire	19
2.4.2 Implémenter l'interface du code de calcul horaire.....	20
2.4.3 Ajouter l'accès aux sorties horaires du calendrier et de la météo	22
2.4.4 exposer une extension du moteur de calcul	24
2.5 Check List : avez-vous bien effectué toutes les étapes ?	25
3. UTILISER L'EXTENSION COUPLEE AU MOTEUR.....	26
3.1 Chargement de l'extension via l'API du moteur RT2012	26
3.2 Chargement de l'extension via le XML des paramètres d'entrée	28
4. STRUCTURE D'UN LIVRABLE.....	30
4.1 Matériel mis à disposition par le CSTB pour le développement	30
4.2 Matériel exigé au développeur de l'extension	30
5. CATALOGUE DES EXTENSIONS	32
5.1 Les générateurs chauffage/Refroidissement/ECS.....	32
5.1.1 A propos des générateurs	32
5.1.2 Implantation dans le jeu de données	32
5.1.3 Implémenter l'interface du jeu de paramètres d'entrée	34
5.1.4 Implémenter l'interface du code de calcul	38
5.1.5 Cas particulier IA pac Double-Service.....	42
5.1.6 Cas particulier le générateur thermodynamique triple-service	42
5.1.7 Cas particulier Les BALLONS Décentralisés d'une production centralisée à appoint décentralisés (PCAD).....	44
5.2 Les Sources Amonts (Des generateurs Thermodynamiques)	49
5.2.1 A propos des Sources Amonts.....	49
5.2.2 Implémentation dans le jeu de Paramètres.....	49
5.2.3 Implementer l'interface du jeu de paramètres d'entrée	50
5.2.4 Implementer l'interface du code de calcul	51
5.3 Les espaces tampons	52
5.3.1 A propos des espaces tampons	52
5.3.2 Implémentation dans le jeu des paramètres	52
5.3.3 Implémenter l'interface du jeu de paramètres d'entrée	53
5.3.4 Implémenter l'interface du code de calcul	54
5.4 la distribution intergroupe.....	59
5.4.1 A propos de la distribution intergroupe	59
5.4.2 La distribution intergroupe ECS.....	60
5.4.3 La distribution intergroupe chaud	64
5.4.4 La distribution intergroupe mixte	69
5.5 Les Productions Stockage.....	72
5.5.1 A propos des productions stockage.....	72
5.5.2 Implémentation dans le jeu de données	72

5.5.3 Implémenter l'interface du jeu de paramètres d'entrée	72
5.5.4 l'interface du code de calcul	74
5.6 Les Ventilations mécaniques	76
5.6.1 A propos des Ventilations mécaniques	76
5.6.2 Implémentation dans le jeu de données	78
5.6.3 Implémenter l'interface du jeu de paramètres d'entrée	79
5.6.4 l'interface du code de calcul	81
5.7 La Gestion et régulation du ballon thermostat.....	84
5.7.1 A propos de la gestion et régulation du ballon Thermostat	84
5.7.2 Implémentation dans le jeu de données	84
5.7.3 Implémenter l'interface du jeu de paramètres d'entrée	85
5.7.4 l'interface du code de calcul	85
5.8 Les Distributions Rafraichissement Direct.....	88
5.8.1 A propos des Distributions Rafraichissement Direct	88
- une au niveau du groupe, Emission_Rafrachissement_Direct, décrite au chapitre 5.9, qui vise à modéliser les émetteurs de rafraichissement et calculer la puissance injectée dans le groupe. Un groupe peut intégrer au maximum une extension de type Rafrachissement_Emission.	88
- une au niveau génération, Distribution_Rafrachissement_Direct, décrite dans le présent chapitre, qui permet d'intégrer la régulation, la distribution entre groupes et les échangeurs de production d'eau froide. L'extension Rafrachissement_Distribution intègre une collection d'extensions Rafrachissement_Emission permettant ainsi de gérer le cas multi-groupes. Elle permet également la communication avec un composant « Source_Amont », afin de récupérer une température de source ainsi que des consommations de circulateurs amont.	88
5.8.2 Implémentation dans le jeu de Paramètres.....	89
5.8.3 Implementer l'interface du jeu de paramètres d'entrée	90
5.8.4 Implementer l'interface du code de calcul	91
5.9 Les Emissions Rafrachissement Direct.....	93
5.9.1 A propos des Emissions Rafrachissement Direct	93
5.9.2 Implémentation dans le jeu de Paramètres.....	93
5.9.3 Implementer l'interface du jeu de paramètres d'entrée	94
5.9.4 Implementer l'interface du code de calcul	95
5.10 Données partagées : Données climatiques	97
5.11 Données partagées : Données temporelles	98
5.12 Méthodes mutualisées accessibles au développement d'une extension dynamique.....	99
5.12.1 Classe RT2012.Extensions.Outils_Prepro_Generateur	99
6. RECOMMANDATIONS DIVERSES:.....	100
6.1 Liste des bibliotheques (DLL) à référencer :	100
6.2 Saisie des données sous forme de tableaux (Matrice) :.....	100
6.2 Jeu de paramètres d'entrée du système titre V:	101
7. ANNEXES.....	103
7.1 Mettre en place un projet d'extension dans Visual Studio 2010	103
7.2 Modèle de fiche algorithme	105
8. MODIFICATIONS	105

1. SPECIFICATIONS GENERALES

1.1 INTRODUCTION

Ce document se décompose en plusieurs parties. La section 1 présente les concepts généraux du mécanisme d' extension dynamique. La section 2 propose de guider le développeur d' une extension dynamique, étape par étape. A noter la section 2.5 qui vous propose un tableau récapitulatif des différentes étapes. La section 3 montre comment utiliser l' extension dynamique couplée au moteur. La section 4 rappelle le contenu d' un livrable dans le cas d' une utilisation Titre V.

Enfin, vous trouverez section 5 l' ensemble des interfaces du moteur, ainsi que la liste des variables climatiques et temporelles disponibles.

1.2 A PROPOS DES EXTENSIONS DYNAMIQUES

Les extensions dynamiques RT2012 permettent de brancher de nouveaux modules, non initialement présent dans la méthode Th-BCE, au cœur de calcul.

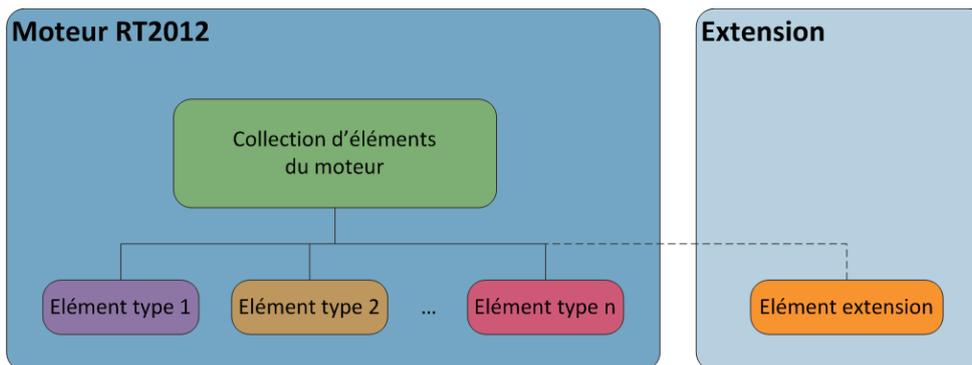


Figure 1 : Schéma des extensions

La solution « extensions dynamiques RT2012 » préserve l' intégrité de la structure de la méthode Th-BCE2012, et permet de brancher dans cette structure prédéfinie de nouveaux modules. Au même titre que les composants initialement prévus dans Th-BCE, les extensions dynamiques devront donc respecter un *contrat* représentant ses interactions avec le reste du moteur.

Remarque : L' extension est utilisée **strictement** comme un élément du moteur. Elle ne peut être utilisée pour raffiner un module existant.

Information : La *Managed Extensibility Framework* version 1 (MEF) est la technologie mise en œuvre pour gérer les extensions. Elle est open-source (<http://mef.codeplex.com/>). Elle prend en charge toute l'instanciation de l'extension et son couplage au moteur suivant des règles implémentées par le CSTB. Ces tâches ne sont pas à effectuer par le développeur de l'extension.

1.3 STRUCTURATION DE L' EXTENSION

Une extension est composée de deux classes :

1. un jeu de paramètres utilisateurs (*paramètres intrinsèques* et *paramètre d' intégration*),
2. un algorithme.

Cette dissociation est nécessaire, le jeu de paramètres étant visible par l' utilisateur final¹, l' algorithme n' étant visible que par le moteur.

L' unique contrainte quant à la technologie informatique utilisée est le développement d' une structure en classe .Net 4.0 (il existe plusieurs éditeurs gratuits de .Net comme MonoDevelop multiplateforme). Ces classes peuvent rester des coquilles vides et communiquer avec n' importe quel langage **sous contrainte d' une compatibilité totale avec Linux, Windows et MACOSX**². Une fois compilées, ces classes forment une dll qui devra être déposée dans un dossier « Extensions » prédéfini du moteur de la RT2012 (voir Figure 2) pour être utilisable par le moteur.

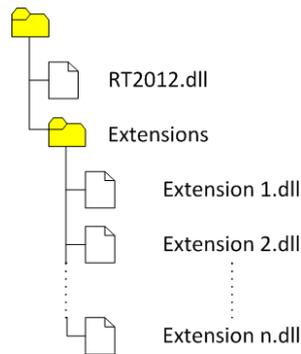


Figure 2 : Arborescence du moteur de calcul RT2012

¹ L' utilisateur final est l' utilisateur du logiciel réglementaire qui embarque le cœur de calcul et les extensions dynamiques. Dans sa phase de test, le développeur de l' extension joue le rôle de l' utilisateur final.

² Les plateformes Smartphone sont exclues du scope.

Les deux classes (celle de paramètres étant notée, pour l' exemple, `Data_Extension` et celle d' algorithme étant notée `Extension`) sont liées l' une à l' autre suivant le schéma UML Figure 3.

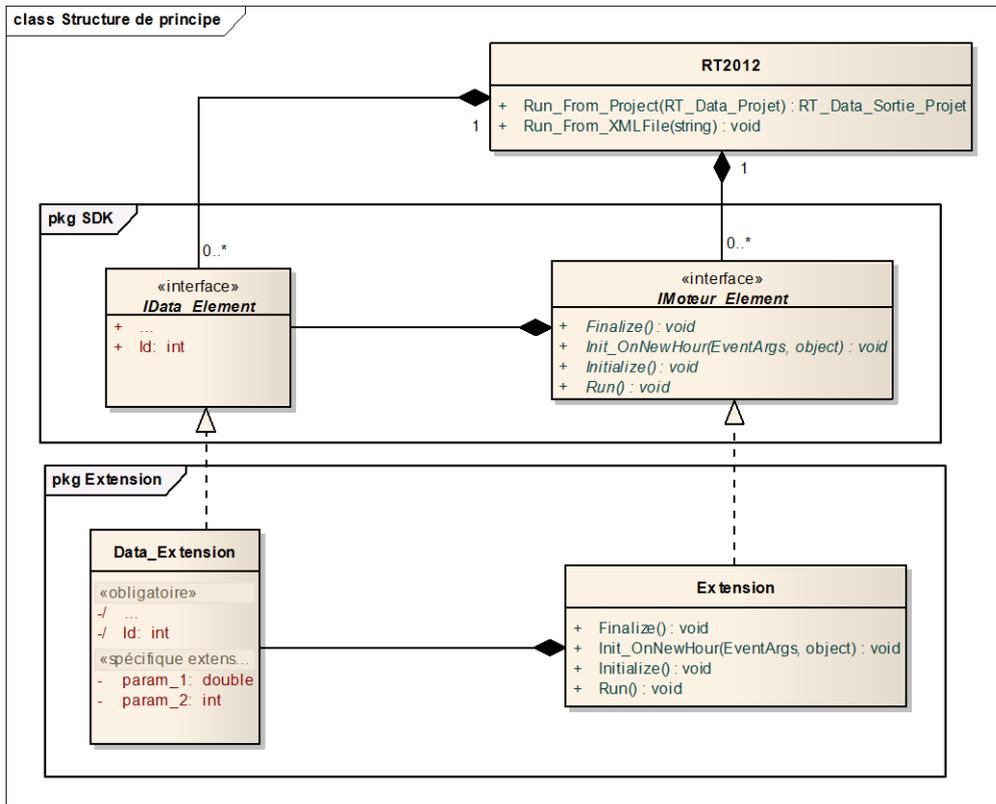


Figure 3: Structure de principe l'extension. Le contenu des interfaces n'est qu'un exemple, et varie d'un type d'extension à l'autre.

Les deux classes à charge du développeur de l' extension (`pkg Extension`) implémentent des interfaces livrées sous forme d' une dll (le package SDK).

La classe `Data_Extension` doit impérativement être serializable en XML. Un XSD des données d' entrée devra être fourni. Les méthodes standard fournies nativement avec les langages .Net peuvent être utilisées.

2. CREER UNE EXTENSION

2.1 REMARQUE LIMINAIRE

La maintenance et le débogage de l' extension dynamique sont assurés par le développeur. En aucun cas le CSTB ne pourra être tenu responsable des bugs présents dans l' extension dynamique.

L' ensemble des méthodes présentées dans cette documentation ont été validées par le CSTB, sauf exceptions dûment mentionnées. Dans ces cas précis, le CSTB ne proposera pas d' aide au développement. Notez que les méthodes utilisées sont toutes largement documentées, et nous incitons le développeur à s' y référer en cas de difficultés.

2.2 CONVENTIONS

Note : dans la suite, tous les éléments devant être personnalisés par le développeur ou dépendant du cas considéré sont précédés d' un symbole « \$ ».

Nom du fichier .dll : L' ensemble de l' interface .Net doit être intégrée dans un unique fichier .dll dont le nom doit impérativement avoir la structure suivante :

T5_ \$NomDeLaSocieteDemandeur_ \$NomDuSysteme .dll

où \$NomDeLaSocieteDemandeur est à remplacer par le nom de la société ou l' entité demandeuse et \$NomDuSysteme et le nom du système Titre V.

Nom d' espace (namespace) : Le namespace de la classe .Net doit être T5_ \$NomDeLaSocieteDemandeur_ \$NomDuSysteme.

Restriction dans les ressources extérieures : L' extension dynamique doit être autoporteuse, en ce sens qu' elle ne doit pas dépendre de matériel tiers que l' utilisateur devrait acquérir par ses propres moyens. Conformément aux exigences de la procédure Titre V, l' ensemble du code source doit pouvoir être mis à disposition de la commission Titre V.

Restriction dans les caractères utilisables : Aucun caractère accentué ou spécial (@, ^,\$, &,#, ~, ..., ç) ne doit être utilisé que ce soit dans les noms de fichiers ou dans le code source.

2.3 CLASSE DE PARAMÈTRES

2.3.1 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES UTILISATEURS

Une extension doit implémenter un jeu de paramètres minimum spécifié par l' interface déclarée dans la librairie RT2012.Extensions.dll correspondant au pkg SDK de la Figure 3. Cette interface est nommée « *IData_ELEMENT* », où *Element* est le type de l' élément qui peut être ajouté au moteur (par exemple *IData_Generateur_Chauffage* pour les paramètres d' un générateur en mode chauffage). L' ensemble de ces interfaces sont mises à disposition via la dll *RT2012.Extensions*³, celle-ci doit donc être référencée par le projet d' extensions dynamiques.

³ L' ensemble des interfaces est documenté dans la section 5

Initialement, la classe de paramètres a la forme suivante :

```
using RT2012.Extensions;

namespace T5_$MaSociete_$MonSysteme
{
    public class $MonExtensionRT2012Data : IData_$Element
    {
    }
}
```

Les déclarations de l'interface doivent être implémentées et les éventuelles propriétés complémentaires du jeu de paramètres d'entrées peuvent être ajoutées. Ces propriétés doivent être sous forme d'accesseurs publiques.

```
using RT2012.Extensions;

namespace T5_$MaSociete_$MonSysteme
{
    public class $MonExtensionRT2012Data : IData_$Element
    {
        #region Membres de l'interface IData_$Element
        public int Index { get; set; }
        public string Name { get; set; }
        #endregion

        #region Membres du jeu de paramètres de l'extension
        public double MaDonneesExtension { get; set; }
        #endregion
    }
}
```

Supprimé: 1

Remarque : les interfaces sur les classes de paramètres n' imposent (en général) que des accesseurs « get » car le moteur n' accède pas en écriture aux paramètres de l' extension. Le setter peut être rajouté si l' utilisateur final doit y avoir accès en édition (ce qui doit être le cas en général).

2.3.2 AJOUTER LA LECTURE DU JEU DE PARAMETRES DEPUIS UN FICHIER XML

Comme précisé dans la section 1.3, la **serialization XML obligatoire**, car ce nouveau jeu de paramètres doit pouvoir être lu depuis le fichier XML d' un projet RT2012. L' environnement .Net est livré avec des méthodes de serialization. Pour les utiliser, des attributs doivent être ajoutés à la classe et éventuellement aux propriétés. Ces attributs proviennent de la bibliothèque « `System.Xml.Serialization` ». Le minimum requis du développeur de l' extension est de spécifier le nom de la balise XML où ces paramètres vont être renseignés, grâce à l' attribut `XmlAttribute`, placé au-dessus de la déclaration de la classe de paramètres.

Par exemple, si le jeu de paramètres d' entrée de l' extension doit être compris entre des balises `<$MonExtensionRT2012></$MonExtensionRT2012>`⁴, la déclaration sera `[XmlAttribute("$MonExtensionRT2012")]`.

La classe ainsi complétée devient

```
using System.Xml.Serialization;
using RT2012.Extensions;

namespace T5_$MaSociete_$MonSysteme
{
    [XmlAttribute("$MonExtensionRT2012")]
    public class $MonExtensionRT2012Data : IData_$Element
    {
        #region Membres de l'interface IData_$Element

        public int Index { get; set; }

        public string Name { get; set; }

        #endregion

        #region Membres du jeu de paramètres de l'extension

        public double MaDonneesExtension { get; set; }

        #endregion
    }
}
```

Optionnellement, chaque accesseur peut être aussi configuré avec des attributs de la librairie « `System.Xml.Serialization` ». La documentation MSDN de cette bibliothèque donne plus d' informations sur ces attributs. Par défaut, les nœuds XML portent le même nom que les propriétés correspondantes (attention à la casse !), et le CSTB conseille de respecter ce comportement par défaut.

⁴ Le nom des balises étant à la charge du développeur, le CSTB conseillant de réutiliser le nom de la classe

2.3.3 EXPOSER UNE EXTENSION DU JEU DE PARAMETRES D' ENTREE

La seule implémentation d' une interface ne suffit pas pour permettre au moteur de reconnaître automatiquement qu' une dll peut être utilisée comme une extension. Des informations doivent y être associées sous la forme d' un nouvel attribut. Cet attribut, défini dans la librairie « `System.ComponentModel.Composition` » est nommé `Export` et il prend en argument le type de l' interface qu' implémente l' extension.

Il est possible d' hériter de plusieurs interfaces. Dans ce cas, il y a toujours une interface « maitresse » qui joue le rôle de donner au moteur le type de système. Les catalogues section 5 mentionnent explicitement l' interface maitresse dans ces cas d' héritage multiple.

```
using System.ComponentModel.Composition;
using System.Xml.Serialization;
using RT2012.Extensions;

namespace T5_MaSociete_MonSysteme
{
    [Export(typeof(IData_$Element))]
    [XmlRoot("$MonExtensionRT2012")]
    public class $MonExtensionRT2012Data : IData_$Element
    {
        #region Membres de l'interface IData_$Element

        public int Index { get; set; }

        public string Name { get; set; }

        #endregion

        #region Membres du jeu de paramètres de l'extension

        public double MaDonneesExtension { get; set; }

        #endregion
    }
}
```

2.3.4 AJOUTER LA DESCRIPTIONS DES PARAMETRES

Une documentation automatique est générée par le CSTB pour chaque élément du jeu de paramètres, et mise à disposition publique entre autres pour les éditeurs de logiciels. Pour ce faire, il est nécessaire de rajouter des attributs aux éléments publics du jeu de donnée comme décrit ci-dessous.

2.3.4.1 Propriétés du jeu de paramètres

Chaque donnée d'entrée, qu'elle soit contrainte par les interfaces ou qu'elle soit spécifique au développement de l'extension dynamique doit être décorée par l'attribut `CSTB_Data_Property` disponible dans la bibliothèque `CSTB_Data_Lib`. Cet attribut permet de préciser

- la définition de la variable par le champ « Title »,
- l'unité par le champ « Unit »,
- la valeur minimale par le champ « Min »,
- la valeur maximale par le champ « Max »,
- la valeur d'affichage par défaut par le champ « Value »,
- si ce champ renvoie à un répertoire par le champ « isDirectory ».

Le champ Title est obligatoire.

```
using System.ComponentModel.Composition;
using System.Xml.Serialization;
using RT2012.Extensions;
using CSTB_Data;

namespace T5_MaSociete_MonSysteme
{
    [Export(typeof(IData_$Element))]
    [XmlRoot("$MonExtensionRT2012")]
    public class $MonExtensionRT2012Data : IData_$Element
    {
        #region Membres de l'interface IData_$Element
        [CSTB_Data_Property(Title = "Index unique")]
        public int Index { get; set; }

        [CSTB_Data_Property(Title = "Nom de mon extension")]
        public string Name { get; set; }

        #endregion

        #region Membres du jeu de paramètres de l'extension
        [CSTB_Data_Property(Title = "Nom de ma donnée", Unit = "kW",
        Value = "0", Min = "0")]
        public double MaDonneesExtension { get; set; }

        #endregion
    }
}
```

2.3.4.2 Cas particulier des tableaux

Si la propriété est un tableau, il est nécessaire d'ajouter l'attribut `CSTB_Data_Matrix` qui définit les légendes des colonnes et des lignes du tableau comme suit :

```
using System.ComponentModel.Composition;
using System.Xml.Serialization;
using RT2012.Extensions;
using CSTB_Data;

namespace T5_MaSociete_MonSysteme
{
    [Export(typeof(IData_$Element))]
    [XmlRoot("$MonExtensionRT2012")]
    public class $MonExtensionRT2012Data : IData_$Element
    {
        #region Membres de l'interface IData_$Element
        [CSTB_Data_Property(Title = "Index unique")]
        public int Index { get; set; }

        [CSTB_Data_Property(Title = "Nom de mon extension")]
        public string Name { get; set; }

        [CSTB_Data_Property(Title = "Ma variable")]
        [CSTB_Data_Matrix(
            RowNumber = 2,
            ColumnNumber = 2,
            RowHeader = "Nom_de_ligne_1 Nom_de_ligne_2",
            ColumnHeader = "Nomcolonne_1 Nomcolonne_2")]
        public double[,] MonTableau { get; set; }

        #endregion

        #region Membres du jeu de paramètres de l'extension
        public double MaDonneesExtension { get; set; }

        #endregion
    }
}
```

2.3.4.3 Cas particulier des « liens »

Les liens permettent de référencer d' autres objets dans le jeu de paramètres. Ceci est par exemple utilisé dans le moteur pour lier les générateurs thermodynamiques avec les sources amont⁵. Ces liens doivent également être décorés dans l' attribut CSTB_Data_Link.

```
using System.ComponentModel.Composition;
using System.Xml.Serialization;
using RT2012.Extensions;
using CSTB_Data;

namespace T5_MaSociete_MonSysteme
{
    [Export(typeof(IData_$Element))]
    [XmlRoot("$MonExtensionRT2012")]
    public class $MonExtensionRT2012Data : IData_$Element
    {
        #region Membres de l'interface IData_$Element
        [CSTB_Data_Property(Title = "Index unique")]
        public int Index { get; set; }

        [CSTB_Data_Property(Title = "Nom de mon extension")]
        public string Name { get; set; }

        [CSTB_Data_Property(Title = "Ma variable")]
        [CSTB_Data_Matrix(
            RowsNumber = 2,
            ColumnsNumber = 2,
            RowsHeader = "Nom_de_ligne_1 Nom_de_ligne_2",
            ColumnsHeader = "Nocolonne_1 Nom_colonne_2")]
        public double[,] MonTableau {get;set}

        [CSTB_Data_Link(Title = "Identifiant de la source amont", Link
= "projet/generation/source_amont")]
        public int Id_Source_Amont { get; set; }

        #endregion

        #region Membres du jeu de paramètres de l'extension
        [CSTB_Data_Property(Title = "Nom de ma donnée", Unit = "kW",
Value = "0", Min = "0")]
        public double MaDonneesExtension { get; set; }

        #endregion
    }
}
```

⁵ Ils jouent un rôle similaire aux IdRef du XML.

2.3.4.4 Déclaration d' un type énuméré pour les listes déroulantes

Il peut être nécessaire de définir des types énumérés, en vue d' implémenter des listes déroulantes dans les interfaces des éditeurs de logiciels. Le développeur de l' extension dynamique à toute liberté pour les définir. S' ils sont utilisés dans le jeu de paramètres, il est également nécessaire de les décorer d' un attribut XmlEnum et CSTB_Data_Enum comme dans l' exemple ci-dessous :

```
using System.ComponentModel.Composition;
using System.Xml.Serialization;
using RT2012.Extensions;

namespace T5_MaSociete_MonSysteme
{
    public enum E_Fonctionnement_Compresseur : int
    {
        [XmlEnum("0")]
        Sans_objet = 0,

        [XmlEnum("1")]
        [CSTB_Data_Enum("Fonctionnement en mode continu du compresseur
ou en cycle marche arrêt du compresseur")]
        Fonctionnement_mode_continu_compresseur = 1,

        [XmlEnum("2")]
        [CSTB_Data_Enum("Fonctionnement en cycle marche arrêt du
compresseur")]
        Fonctionnement_cycle_marche_arret_compresseur = 2
    }
}
```

2.3.5 GENERATION DU XSD

Le XSD (XML Schema Definition) est une définition standardisée d' un XML. Le XSD du jeu de paramètres doit être impérativement livré avec l' extension dynamique. Il sert de base aux éditeurs de logiciel pour valider la structure du jeu de paramètres

Le CSTB préconise la méthode suivante pour générer le XSD. Le SDK du .Net est livré avec l' outil xsd.exe. En général, cet exécutable est disponible dans un dossier du type C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin\NETFX 4.0 Tools\x64. Le répertoire varie suivant les installations.

Le code doit être compilé (ce qui à ce niveau du développement doit être possible), et l'outil xsd.exe doit être utilisé sur la dll dans une invite de commande Windows comme suit :

```
#xsd.exe T5_$NomDeLaSocieteDemandeur_$NomDuSysteme.dll /type:$MonExtensionRT2012Data
```

Cette commande va générer un fichier schema0.xsd à joindre au dossier, de préférence à renommer avec le nom de l'extension.

2.4 L' ALGORITHME

2.4.1 A PROPOS DU CODE DE CALCUL HORAIRE

Le calcul de la RT2012 est basé sur un enchaînement séquentiel de calculs effectué à chaque pas de temps (horaire). Une extension permet d' implémenter un élément de la séquence. A chaque pas de temps, le moteur a besoin de connaître la valeur des sorties d' une extension. Le moteur s' occupe d' affecter les valeurs des variables d' entrée prédéfinies de l' extension et appelle des méthodes de calculs de sorties prédéfinies. Après l' appel à une méthode de calcul, le moteur récupère certaines valeurs en sortie.

A aucun moment le développeur de l' extension ne prend en charge les appels de l' extension par le moteur. Il ne fait qu' implémenter le contenu de méthodes prédéfinies dans les interfaces, le moteur s' occupant seul de faire les appels.

Les rajouts de méthodes par rapport aux méthodes prédéfinie est possible, mais celles-ci ne pourront pas interagir avec le moteur.

Suivant les extensions, ce code de calcul peut être réparti dans différentes méthodes prédéfinies, appelées par le moteur dans un ordre prédéfini. Par défaut, on trouvera la méthode « `Initialise` » qui est appelée au début de la simulation, la méthode « `Termine` » qui est appelée à la fin de la simulation, et la méthode `Init_OnNewHour` appelée au début de chaque pas de temps systématiquement.

2.4.2 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL HORAIRE

Pour ajouter l' algorithme qui représente le fonctionnement horaire de l' extension, une classe qui implémente l' interface du type de l' extension doit être créée, comme pour le jeu de paramètres d' entrée.

```
namespace T5_$MaSociete_$MonSysteme
{
    public class $MonExtensionRT2012 : IMoteur_$Element
    {
        #region IMoteur_$Element Membres

        public IData_Element Data { get; set; }

        public void Initialise(object sender, System.EventArgs e)
        {
            // implémentation de la méthode d'initialisation
        }

        public void Termine(object sender, System.EventArgs e)
        {
            // Implémentation de la méthode de fin de simulation
        }

        // Liste des propriétés et méthodes déclarée dans l'interface
        #endregion
    }
}
```

Parmi les propriétés déclarées dans l' interface, la propriété nommée « Data » contient l' instance de la classe du jeu de paramètres d' entrée remplie avec les valeurs qui proviennent du projet à simuler. Cette propriété n' est affectée qu' une fois la méthode « Initialise » appelée par le moteur. Elle n' est donc pas disponible dans le constructeur, à ce niveau tout accès à cette propriété renverra null.

On retrouve ensuite les méthodes « Initialise » et « Termine » et « Termine ».

Initialise : appelée à la suite de la première construction

Termine : appelée à la fin du calcul

Init_OnNewHour : appelée au début de chaque pas de temps.

Du point de vue de l' interface, les variables d' entrée sont accessibles en lecture et écriture (get, set), et les sorties en lecture seule (get uniquement). Bien évidemment, un

setter peut être implémenté dans la classe d' extension, mis il ne sera pas visible du moteur (le moteur ne peut pas changer les valeurs des variables de sortie d' une extension).

Les autres méthodes déclarées dans l' interface doivent être implémentées, et seront appelées par le moteur.

2.4.3 AJOUTER L' ACCES AUX SORTIES HORAIRES DU CALENDRIER ET DE LA METEO

Le concepteur de l' extension peut, en plus des entrées horaires prévues par le type de l' extension, se connecter à des sorties provenant du calendrier et de la météo (la liste des sorties horaires du calendrier et de la météo est documentée sections 0 et 5.11). Pour se connecter à ces sorties, des propriétés de type « *ISortieMoteur<>* » sont nécessaires. C' est un type générique qui prend le type de la donnée horaire comme identifiant (le type de chaque sortie est spécifié dans l' annexe sur les sorties du calendrier et de la météo) et encapsule sa valeur accessible en lecture seule. A titre d' exemple, la sortie horaire « *T_simul* » du calendrier devient accessible dans l' extension quand une propriété de type « *ISortieMoteur<int>* » est déclarée. Il suffit d' attribuer à cette propriété un attribut de type « *RT_SortieCalendrier* » (ou, dans le cas d' une sortie climatique un attribut « *RT_SortieMeteo* »). Ces deux attributs prennent en argument une valeur des énumérations « *RT_Types_SortiesCalendrier* » et « *RT_Types_SortiesMeteo* », qui contiennent toutes les sorties calendrier et météo disponibles

```
using System.ComponentModel.Composition;
using RT2012.Extensions;

namespace T5_$MaSociete_$MonSysteme
{
    public class $MonExtensionRT2012 : IMoteur_$Element
    {
        #region Implémentation IRT_$Element

        ...

        #endregion

        #region Autres entrées horaires

        [RT_SortieCalendrier(RT_Types_SortiesCalendrier.T_Simul)]
        public ISortieMoteur<int> TempsSimulation { get; set; }

        [RT_SortieClimat(RT_Types_SortiesClimat.Vent)]
        public ISortieMoteur<double> VitesseVent { get; set; }

        #endregion

    }
}
```

La propriété « Valeur » de « ISortieMoteur<> » (par exemple, TempsSimulation.Valeur) permet d' accéder à la valeur numérique de la propriété. En cas d' erreur sur la connexion de l' entrée horaire de l' extension et la propriété issue du calendrier ou de la météo, l' entrée horaire de l' extension sera affectée à nu11. Cet état peut être vérifié au moment de l' initialisation.

2.4.4 EXPOSER UNE EXTENSION DU MOTEUR DE CALCUL

Pour que cette classe, représentant les calculs horaires de l' extension, soit reconnue par le moteur, deux attributs provenant de la librairie « System.ComponentModel.Composition » doivent être ajoutés. Le premier attribut est le même que pour le jeu de paramètres d' entrée « Export ». Cette fois-ci le paramètre sera le nom de l' interface du type de l' extension. Le deuxième attribut est « ExportMetadata », il va permettre de donner des informations concernant cette extension. L' information requise par l' extension est « DataType » qui permet de spécifier le type de la classe qui contient le jeu de paramètres d' entrée de l' extension⁶.

```
using System.ComponentModel.Composition;
using RT2012.Extensions;

namespace T5_$MaSociete_$MonSysteme
{
    [Export(typeof(IMoteur_$Element))]
    [ExportMetadata("DataType", "
T5_$MaSociete_$MonSysteme.$MonExtensionRT2012Data")]
    public class $MonExtensionRT2012 : IMoteur_$Element
    {
        #region IMoteur_$Element Membres

        public IData_$Element Data { get; set; }

        public void Initialise(object sender, System.EventArgs e)
        {
            // implémentation de la méthode d'initialisation
        }

        public void Termine(object sender, System.EventArgs e)
        {
            // Implémentation de la méthode de fin de simulation
        }

        // Liste des propriétés et méthodes déclarée dans l'interface

        #endregion
    }
}
```

⁶ Cet attribut permet au MEF de créer le lien entre le jeu de données et la classe de paramètres.

2.5 CHECK LIST : AVEZ-VOUS BIEN EFFECTUE TOUTES LES ETAPES ?

Nous vous proposons un tableau récapitulatif des actions à mener lors du développement d' une extension dynamique. Celui-ci peut vous servir de base pour vérifier que vous êtes passés par toutes les étapes

	Action	Section du document	
	Liste des interfaces dont j' ai besoin (voir section 5)	5	<input type="checkbox"/>
	Documentation Fiche algorithme basée sur le modèle fourni	7.2	<input type="checkbox"/>
Classe de paramètres	Création de la classe donnée	2.3.1	<input type="checkbox"/>
	Héritage des interfaces de paramètres	2.3.1	<input type="checkbox"/>
	Implémentation des paramètres hérités	2.3.1	<input type="checkbox"/>
	Rajout des paramètres spécifiques à l' extension	2.3.1	<input type="checkbox"/>
	Ajouter la lecture/écriture XML	2.3.2	<input type="checkbox"/>
	Exposer le jeu de donnée	2.3.3	<input type="checkbox"/>
	Décoration des propriétés et des types énumérés avec les CSTB_Data_XX et XMLEnum	2.3.4	<input type="checkbox"/>
	Génération du XSD	2.3.5	<input type="checkbox"/>
Classe algorithme	Création de la classe de calcul	2.4.2	<input type="checkbox"/>
	Héritage des interfaces de calculs	2.4.2	<input type="checkbox"/>
	Implémenter les méthodes héritées	2.4.2	<input type="checkbox"/>
	Exposer la classe de calcul	0	<input type="checkbox"/>
	Effectuer un test unitaire		<input type="checkbox"/>
	Effectuer les tests d' intégration au moteur		<input type="checkbox"/>

3. UTILISER L' EXTENSION COUPLEE AU MOTEUR

La section 1.3 donne la structure de classe. Une fois l' extension implémentée en suivant la section 2, nous montrons comment effectuer l' appel de l' extension dynamique par le moteur. Deux solutions existantes.

1. Soit par l' API, la présence du Data_Extension.
2. Soit par le XML des paramètres d' entrée

Quelle que soit la solution utilisée, l' utilisateur final ne devra que travailler avec le jeu de paramètres. A aucun moment la classe de calcul n' est manipulée par l' utilisateur final, le moteur s' occupe lui-même de cette partie. Utiliser une extension revient donc à utiliser la classe de paramètres.

Cette étape est détaillée pour que le développeur puisse effectuer des tests. Il n' est pas nécessaire de posséder un logiciel réglementaire. Le développeur devra faire appel au moteur, par exemple en se basant sur un fichier XML préexistant.

3.1 CHARGEMENT DE L' EXTENSION VIA L' API DU MOTEUR RT2012

Cette approche est détaillée ici pour les éditeurs de logiciels. Le CSTB ne propose pas cette approche pour effectuer des tests, celle détaillée section 3.2 par le XML d' entrée est plus facile d' utilisation.

Cette solution nécessite que le logiciel référence l' extension dynamique pour pouvoir l' instancier. On présente Figure 4 la séquence de principe du couplage entre le moteur et les extensions.

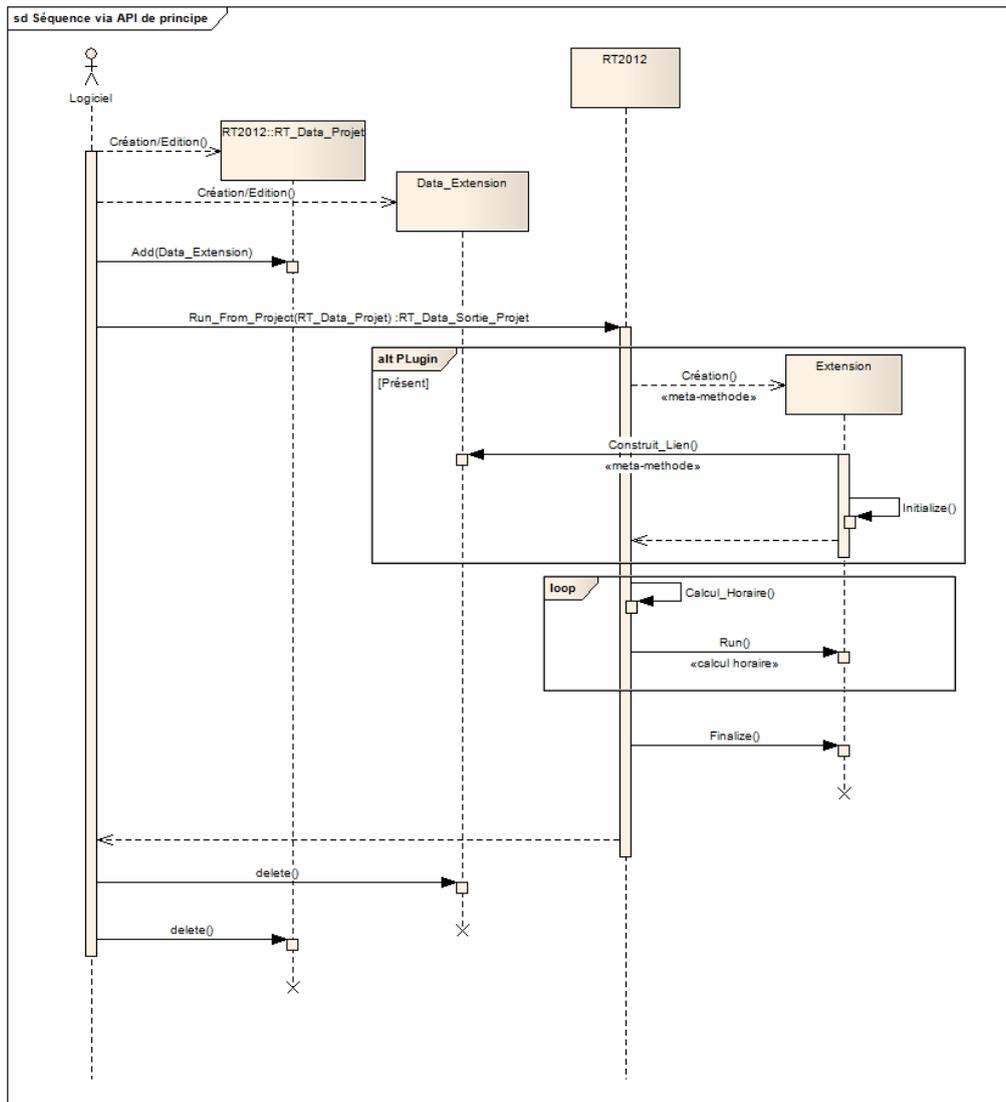


Figure 4 : Diagramme de séquence de principe lorsqu'une d'une configuration du moteur via les API

Comme on peut le voir Figure 4, la seule action effectuée par l' utilisateur final, ici matérialisé par le logiciel utilisant le moteur de calcul, est l' instantiation de la classe de paramètres. L' interaction entre le moteur et la classe de calcul de l' extension est gérée automatiquement. Seules les actions d' édition/création de la classe de paramètres, et son ajout au jeu de paramètres initial sont nécessaires.

3.2 CHARGEMENT DE L' EXTENSION VIA LE XML DES PARAMETRES D' ENTREE

Cette solution ne nécessite pas que le logiciel référence l' extension dynamique ; le moteur effectuera l' instantiation via les mécanismes du MEF lorsqu' une extension dynamique est découverte dans le fichier XML des paramètres d' entrée.

On présente Figure 5 la séquence de principe du couplage entre le moteur et les extensions

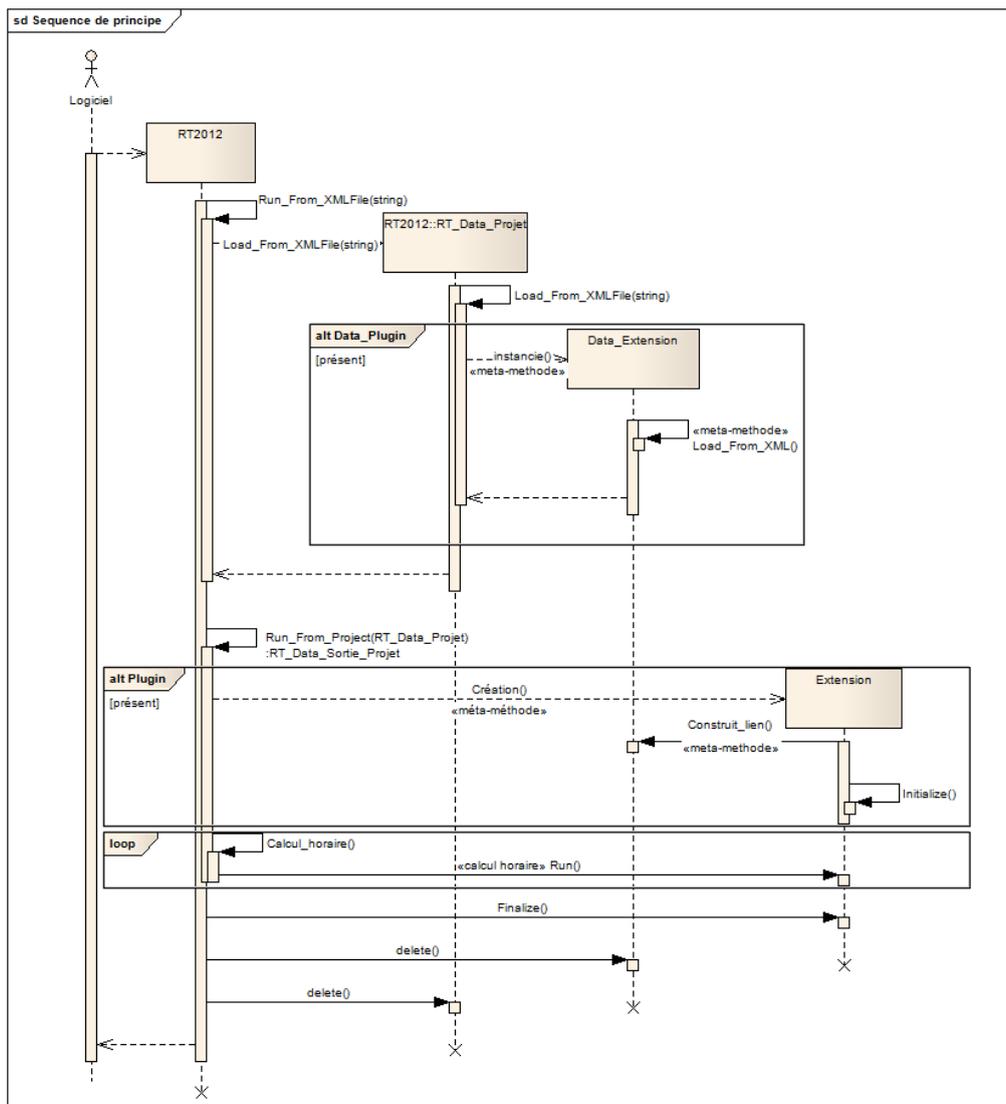


Figure 5 : Diagramme de séquence de la prise en compte des extensions dynamiques. Les méthodes notées « meta-méthode » sont automatiquement prises en charge par l'environnement MEF.

Dans le cas d' une configuration via le fichier XML du jeu de paramètres, les paramètres seront placés dans un nœud XML définissant l' extension (voir section 2.3.2). Ce nœud est à intégrer dans la collection des éléments du même type que l' extension. Voici un exemple d' insertion des paramètres d' une extension dans une collection de générateurs :

```
<Generateur_Collection>
  <Generateur_Effet_Joule>
    <Id>0</Id>
    <Name>Mon generateur effet joule</Name>
    ...
  </Generateur_Effet_Joule>
  ...
  <$Generateur_Extension>
    <Id>6</id>
    <Name>Mon generateur extension</Name>
    ...
  </$Generateur_Extension>
</Generateur_Collection>
```

Le nœud `$Generateur_extension` est à remplacer par le contenu de XmlRoot défini dans la section 2.3.4.

4. STRUCTURE D' UN LIVRABLE

4.1 MATERIEL MIS A DISPOSITION PAR LE CSTB POUR LE DEVELOPPEMENT

- La bibliothèque de calcul RT2012 sous forme de la dll RT2012.dll
- Le SDK de création des extensions sous forme de la dll RT2012.Extensions
- La collection des propriétés des données dans la dll CSTB_Data_Lib (nécessaire pour la section 2.3.4)
- Cette documentation
- Un modèle de fiche algorithme
- Un exemple de développement (sous forme d' un projet Visual Studio 2010)

4.2 MATERIEL EXIGE AU DEVELOPPEUR DE L' EXTENSION

Remarque : Cette section ne remplace pas la liste des documents nécessaires à l' établissement d' un Titre V. Elle précise uniquement les éléments accompagnant la dll de l' extension dynamique.

L' extension dynamique se présente sous la forme d' une dll développée suivant les indications des sections précédentes (et éventuellement des dll auxiliaires, sous la contrainte de la section 2.2). **Sont également exigés** :

- Un XSD de la représentation XML des paramètres utilisateur
- Des cas d' application (moteur RT2012 + paramétrage de l' extension) accompagnés des jeux de données de sortie du moteur RT2012 correspondant.
- Une documentation sous la forme d' une fiche algorithme. La fiche algorithme devra mentionner les tests. Chacun des tests mentionnés dans le point ci-dessus devra être décrit dans la fiche algorithme.
- Exposition COM avec son TLB pour garantir la compatibilité avec les langages anciens sous Windows.
- Les variables du code doivent être strictement identiques à la documentation
- Numéro de version dans la configuration de la dll.

- Signer avec nom fort.
- Version privée : en mode Debug avec son fichier .pdb

5. CATALOGUE DES EXTENSIONS

5.1 LES GENERATEURS CHAUFFAGE/REFROIDISSEMENT/ECS

5.1.1 A PROPOS DES GENERATEURS

Un générateur est un système utilisé pour fournir de l'énergie afin de répondre aux besoins des bâtiments en chauffage, refroidissement et eau chaude sanitaire (ECS). A chaque pas de temps de la simulation, le moteur va demander au générateur une puissance requise au générateur en chaud, froid et ECS, selon les fonctions du générateur. Le générateur répondra à cette demande selon ses capacités en retournant la puissance qu'il a pu fournir.

5.1.2 IMPLANTATION DANS LE JEU DE DONNEES

Remarque : une bonne connaissance de la structure des données du moteur de calcul est nécessaire pour la suite.

Des générateurs peuvent être rajoutés dans la collection `RT_Data_Generateur_Collection` ou dans la collection `RT_Data_Source_Ballon_Base_Collection`.

5.1.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

	Instantané chauffage	Instantané / ECS	Instantané Thermo Chauffage / ECS	Pour stockage ECS / Chauffage Thermo	Source Ballon Boucle solaire ECS / Chauffage
IData_Service_Chauffage	X	X	X	X	X
IData_Service_Refroidissement					
IData_Service_ECS		X	X	X	X
IData_Generateur_Thermodynamique			X	X	
IData_Generateur_Instantane_Extension	X	X	X		
IData_Source_Ballon_Base_Extension				X	X
IData_Boucle_Solaire					X

Tableau 1 : Combinaison des interfaces pour la partie paramètre des générateurs

Le jeu de paramètres d' entrée d' un générateur dépend de la nature du générateur. Deux interfaces maitresses **mutuellement exclusives**⁷

- Pour un générateur instantané, implémenter l' interface
« **IData_Generateur_Instantane_Extension** »
- Pour un générateur de base pour un ballon, implémenter l' interface
« **IData_Source_Ballon_Base_Extension** »

Les paramètres associés à ces deux interfaces sont identiques :

Nom	Description	Unité	Type	Intervalle
Name	Nom du générateur	-	String	-
Index	Identifiant unique d'un générateur	-	int	[0 ; +∞]
Rdim	Nombre de générateurs identiques	-	int	[1 ; +∞]
Id_Fl_Aval	Identificateur du fluide aval Eau(1) Air(2) Sol(3)	-	E_Id_Fluide	[1 ; 3]
Id_Fou_Gen	Fonction du générateur (Ch, Fr, Ecs, Ch+Ecs ou Ch+Fr)	-	RT_Types_ Fonctionnements_Generateur	

⁷ Pour rappel, il est impossible d'avoir deux interfaces maitresses, celles-ci étant utilisées dans l'export détaillé section 2.3.3.

Suivant les services, les trois interfaces suivantes peuvent être rajoutées et ne sont pas exclusives:

1. **IData_Service_Chauffage**

Nom	Description	Unit	Type	Intervalle
Name	Nom du générateur	-	String	-
Index	Identifiant unique d'un générateur	-	int	[0 ; +∞]
Rdim	Nombre de générateurs identiques	-	int	[1 ; +∞]
Idpriorite_Ch	Indice de priorité du générateur en chauffage	-	int	[1 ; +∞]
Theta_max_av_IGen	Température avale maximale	-	double	

2. **IData_Service_Refrondissement:**

Nom	Description	Unité	Type	Intervalle
Name	Nom du générateur	-	String	-
Index	Identifiant unique d'un générateur	-	int	[0 ; +∞]
Rdim	Nombre de générateurs identiques	-	int	[1 ; +∞]
Idpriorite_Fr	Indice de priorité du générateur en refroidissement	-	int	[1 ; +∞]
Theta_min_av_IGe n	Température avale minimale	-	double	

3. **IData_Service_ECS (non compatible avec un stockage):**

Nom	Description	Unité	Type	Intervalle
Name	Nom du générateur	-	String	-
Index	Identifiant unique d'un générateur	-	int	[0 ; +∞]
Rdim	Nombre de générateurs identiques	-	int	[1 ; +∞]
Idpriorite_Ecs	Indice de priorité du générateur en ECS	-	int	[1 ; +∞]

Ensuite, dans le cas d'un générateur thermodynamique, l'interface **IData_Generateur_Thermodynamique** doit être rajoutée et contient les paramètres suivants :

IData_Generateur_Thermodynamique				
Nom	Description	Unité	Type	Intervalle
Id_Fl_Amont	Identificateur du fluide amont Eau(1) Air(2) Sol(3)	-	E_Id_Fluide	[1 ; 3]
Id_Source_Amont	Identifiant de la source amont	-	int	

Remarque : lorsque les paramètres apparaissent en double voire triple, (si plusieurs services sont sélectionnés par exemple), ils ne doivent être implémentés qu'une seule et unique fois.

Ensuite, dans le cas d' un source ballon base sur boucle solaire,, l' interface `IData_Boucle_Solaire` doit être rajoutée et contient les paramètres suivants :

IData_Generateur_Thermodynamique				
Nom	Description	Unité	Type	Intervalle
Alpha	Azimut (0° : Sud ; 90° : Ouest ; 180° : Nord ; 270° : Est)	°	double	[0 ; 360[
Beta	Inclinaison de 0° (horizontale vers le haut) à 90° (verticale)	°	double	[0 ;90]
Masque_Collection*	Collection des masques proches	-	Data_Masque_Collection	
Is_regulateur_temperature	Présence d'un régulateur sur la température. Sinon, c'est sur le rayonnement	-	bool	
COMETH_Masque_Collection***	Référence sur la collection des masques dans le jeu de données d'entrée de COMETH		ICOMETH_Data_Masque_Collection	
Id_Mas **	Id_Mas: définit un accesseur get et set, pour l'indice des masques(-)		int[]	
Id_Ori**	Indicateur de l'orientation		int	[0, +∞]

(*) : La collection *Data_Masque_Collection* est une collection de jeu donnée d' entrée des masques proches telles qu' elles sont conçues dans la RT2012. Cette collection a été rendu disponible aux développeurs d' extension dynamique de type « boucle solaire ». Elle est située dans l' espace de noms « *RT2012.Extensions.PlugOut* ».

(**)Ces paramètres sont remplis par le moteur. Par conséquent, il faut lors de leur implémentation, ajouter le décorateur [*XmlIgnore*] pour qu' ils ne soient pas pris en compte par le Sérialiseur Xml.

(***) Le paramètre *COMETH_Masque_Collection* est une liste de composants d'entrée des masques internes au moteur de calcul de la *RT2012* appelé « *ICOMETH_Data_Masque* ». Cette liste est automatiquement remplie par le moteur sur la base du paramètre d'entrée précédent (*Data_Masque_Collection*) par un préprocesseur interne au moteur. Elle n'est donc pas remplie par l'utilisateur. Par conséquent, il faut lors de son implémentation, ajouter le décorateur [*XmlIgnore*] pour qu'elle ne soit pas prise en compte par le Sérialiseur Xml.

5.1.4 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

	Instantané chauffage	Instantané / ECS	Thermo Chauffage / ECS	Pour stockage ECS / Chauffage Thermo	Source Ballon Boucle solaire ECS / Chauffage
IMoteur_Service_Chauffage	X	X	X	X	x
IMoteur_Service_Refroidissement					
IMoteur_Service_ECS		X	X	X	x
IMoteur_Generateur_Thermodynamique			X	X	
IMoteur_Generateur_Extension	X	X	X	X	x
IMoteur_Boucle_Solaire					x
IMoteur_Service_Chauffage	X	X	X	X	x

Tableau 2 : Combinaison des interfaces pour la partie algorithme du générateur

La partie algorithme du générateur extension dynamique hérite toujours de *IMoteur_Generateur_Extension*. Son contenu en variable est le suivant :

Entrées du composant

Nom	Description	Unité	Type	Intervalle
Theta_amb	Température d'ambiance du lieu où se trouve le générateur	°C	double	[-∞ ; +∞]

Sorties

Nom	Description	Unité	Type	Intervalle
Id_Fou_Gen	Fonctionnement du générateur	-	RT_Types_Fonctionnements_Generateur	-
Qrest	Energie restant à fournir à la fin du pas de temps, faisant l'objet d'un report de demande à un autre générateur en séquence ou au pas de temps suivant	Wh	double	[0 ; +∞]
Phi_vc	Pertes thermiques et puissances des auxiliaires du générateur transmises vers l'ambiance	Wh	double	[0 ; +∞]

Qprelec	Production électrique du générateur	Wh	double	[0 ; +∞]
Waux_pro	Consommation des auxiliaires au pas de temps h	Wh	double	[0 ; +∞]
Pmax	Puissance maximale du générateur	W	double	[0 ; +∞]
Q_cons	Puissance consommée par le générateur	Wh	double	[0 ; +∞]
Taux_Charge	Taux de charge du générateur	-	double	[0 ; 1]
Q_fou	Puissance fournie par le générateur	W	double	[0 ; +∞]

Suivant les services, une ou plusieurs des trois interfaces suivantes doivent être rajoutées et ne sont pas exclusives :

1. IMoteur_Service_Chauffage

Entrées du composant

Nom	Description	Unité	Type	Intervalle
-----	-------------	-------	------	------------

Sorties

Nom	Description	Unité
-----	-------------	-------

Pn_Gen_Ch	Puissance nominale du générateur en chauffage	W	Double	[0 ; +∞]
-----------	---	---	--------	----------

Q_Cef_Ch	Tableau des consommations en énergie finale pour le chauffage en fonction des énergies utilisées (Gaz, Fioul, Charbon, Bois, Electricité, Réseau de fourniture)	Wh	RT_Consommations_Energies	[0 ; +∞]
----------	---	----	---------------------------	----------

2. IMoteur_Service_Refrondissement

Entrées du composant

Nom	Description	Unité	Type	Intervalle
-----	-------------	-------	------	------------

Id_Refrroidi	Indication de l'appel du Générateur en refroidissement	-	int	[1 ; +∞]
--------------	--	---	-----	----------

Sorties

Nom	Description	Unité
-----	-------------	-------

Pn_Gen_Fr	Puissance nominale du générateur en refroidissement	W	Double	[0 ; +∞]
-----------	---	---	--------	----------

Q_Cef_Fr	Tableau des consommations en énergie finale	Wh	RT_Consomma	[0 ; +∞]
----------	---	----	-------------	----------

pour le refroidissement en fonction des énergies utilisées (Gaz, Fioul, Charbon, Bois, Electricité, Réseau de fourniture)

tions_
Energies

3. IMoteur_Service_ECS

Entrées du composant

Nom	Description	Unité	Type	Intervalle
IdEcs_Seu	Indication de production ECS seule	-	int	[1 ; +∞]

Sorties

Nom	Description	Unité	Type	Intervalle
R_fonctecs	Temps de fonctionnement du générateur pour la production d'ECS, à puissance maximale, au pas de temps h	-	Double	[0 ; 1]
Pn_Gen_Ecs	Puissance nominale du générateur en ECS	W	Double	[0 ; +∞]
Q_Cef_Ecs	Tableau des consommations en énergie finale pour le refroidissement en fonction des énergies utilisées (Gaz, Fioul, Charbon, Bois, Electricité, Réseau de fourniture)	Wh	RT_Consommations_ Energies	[0 ; +∞]

Pour ces trois interfaces, on est tenu d' en implémenter au moins une, tout en étant en cohérence avec la valeur d' *Id_Fou_Gen* définie en paramètre (Voir paragraphe précédent).

Chacune de ces trois interfaces déclare une méthode nommée « Run_Ch » pour le chauffage, « Run_Fr » pour le refroidissement et « Run_Ecs » pour l' ECS. Chacune de ces méthodes prend les paramètres suivant :

Nom	Description	Unité	Type	Intervalle
aQreq	Energie requise en entrée du générateur	Wh	double	[0 ; +∞]
aThetaAval	Température avale	°C	double	[-∞ ; +∞]
aRat_puiDispo	Ratio de puissance disponible	-	double	[0 ; 1]

Ces méthodes retournent la puissance fournie par le générateur :

Nom	Description	Unité	Type	Intervalle
-	Energie totale effectivement fournie par le générateur au pas de temps h	Wh	double	[0 ; +∞]

Le contenu des méthodes doit comprendre tout le calcul qui permet d'obtenir l'énergie totale fournie en mode chauffage, refroidissement ou ECS. A l'issue de l'appel la méthode, toutes les sorties de chaque interface doit avoir été calculée.

Dans le cas d'un source ballon sur boucle solaire, l'interface **IMoteur_Boucle_Solaire** suivante doit être rajoutée et contient les données suivantes :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Isr_star	Irradiance (ensoleillement) sur le plan des capteurs	W/m ²	double	[0 ; +∞]
Id_Mas	Id_Mas: définit un accesseur get et set, pour l'indice des masques(-)	-	int[]	
Id_Ori	Indicateur de l'orientation.	-	int	[0, +∞]
Masque_Collection	Référence sur la collection des masques dans le moteur COMETH	ICOMETH_Masque_Collection		

Sorties				
Nom	Description	Unité	Type	Intervalle
P_sol	Puissance transférée à l'échangeur	W	Double	[0 ; +∞]
P_p	Puissance consommée par la pompe	W	Double	[0 ; +∞]

Enfin, dans le cas d'un générateur thermodynamique, l'interface **IMoteur_Generateur_Thermodynamique** suivante doit être rajoutée et contient les données suivantes :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Source_Amont	Référence vers la source amont du générateur thermodynamique	-	IMoteur_Source_Amont	[1 ; +∞]
Theta_amont	Température amont dans le cas des générateurs thermodynamiques	°C	double	[-50 ; +∞]
Waux_Amont	Consommations électrique des auxiliaires amont d'un générateur	Wh	double	[0 ; +∞]

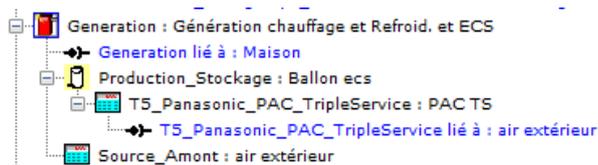
Sorties				
Nom	Description	Unité	Type	Intervalle
Phi_Rejet	Puissance nominale du générateur en chauffage	W	Double	[0 ; +∞]

5.1.5 CAS PARTICULIER LA PAC DOUBLE-SERVICE

5.1.6 CAS PARTICULIER LE GENERATEUR THERMODYNAMIQUE TRIPLE-SERVICE

5.1.6.1 Assemblage d'un générateur thermodynamique triple service

Pour fonctionner correctement le générateur doit être placé dans une production de stockage de type Base sans appoint.



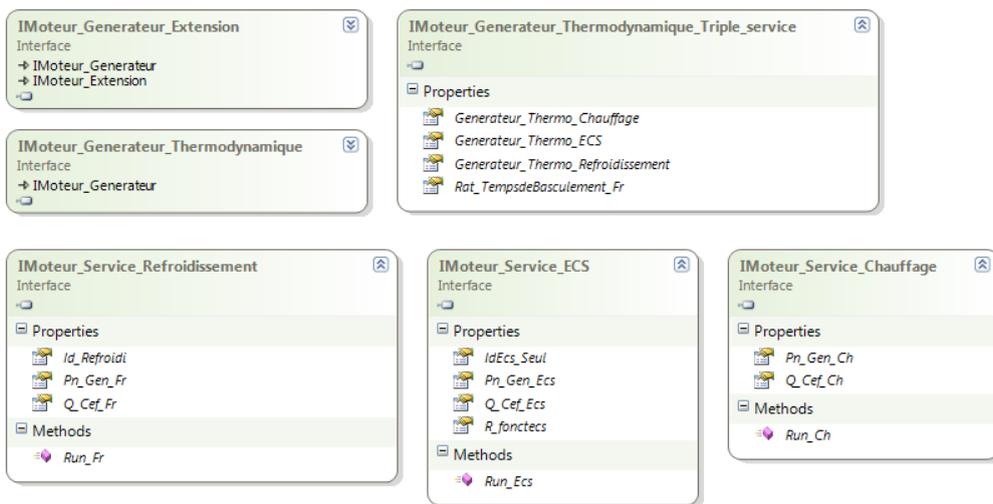
5.1.6.2 Implementation des interfaces de données

Le générateur thermodynamique triple service, doit hériter des interfaces suivantes :

- IData_Source_Ballon_Base_Extension

- IData_Generateur_Thermodynamique
- IData_Service_ECS
- IData_Service_Chauffage
- IData_Service_Refrondissement

5.1.6.3 Implémentation des interfaces du code de calcul

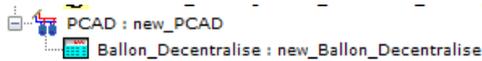


L' interface `IMoteur_generateur_thermodynamique_Triple_service` doit être rajoutée et contient les données suivantes :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Generateur_Thermo_Chauffage	La partie du générateur faisant du chaud		IMoteur_Generateur_Thermodynamique	
Generateur_Thermo_Refrondissement	la partie du générateur faisant de froid		IMoteur_Generateur_Thermodynamique	
Generateur_Thermo_ECS	la partie du générateur faisant de Ecs		IMoteur_Generateur_Thermodynamique	

Rat_TempsdeBasculement_Fr	Ratio équivalent au Temps de basculement en froid dans l'heure	-	Double	[0 ; 1]
---------------------------	--	---	--------	---------

5.1.7 CAS PARTICULIER LES BALLONS DECENTRALISES D'UNE PRODUCTION CENTRALISEE A APPOINT DECENTRALISES (PCAD)



5.1.7.1 Implementation des interfaces de données



L'extension doit implémenter l'interface « IRT_Data_Ballon_Decimalise » qui hérite de 3 autres interfaces voir schéma ci-dessus.

Données :

5.1.7.1.1 Interface ICOMETH_Data_Generateur :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Index	Indice unique du générateur	-	int	
Name	Nom du generateur		string	
Rdim	Nb de générateur(s) identique	-	int	
Id_FI_Aval	Identificateur du fluide de aval Eau(1) Air(2) Fluide(3)	-	int	
Id_Fou_Gen	Fonction du générateur	-	int	

5.1.7.1.2 Interface ICOMETH_Data_Assemblage_Ballon_Decentralise_PCAD

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
nb_sto_b	Nombre d'élément de stockage d'ECS strictement identiques connectés à la même distribution primaire	-	int	
Distribution_Intergroupe_ECS	référence vers la distribution intergroupe en ECS	-	ICOMETH_Distribution_Intergroupe_ECS	
Distribution_Intergroupe_Chaut	référence vers la distribution intergroupe en Chaut		ICOMETH_Distribution_Intergroupe_Chaut	
Assemblage_Ballon	référence vers le ballon base+appoint		ICOMETH_Assemblage_Ballon	

5.1.7.1.3 Interface IRT_Data_Ballon_Decentralise

Entrées du composant				
Nom	Description	Unité	Type	Intervalle

Id_Fou_Gen	Fonction du générateur	RT_Types_Fonctionnements_Generateur	int
Source_Ballon_Appoint_Collection	Générateurs d'appoint du ballon	-	Data_Source_Ballon_Appoint_Collection
Type_prod_stockage	Type de stockage	-	RT_Type_Ballon
nb_sto_b	Nombre d'éléments de stockage d'ECS strictement identiques	-	Int
nb_assembl	Nombre d'assemblages strictement identiques à considérer au niveau de la génération	-	Int
V_tot	Volume total du ballon	-	double
Statut_faux	Statut de faux	-	RT_Valeur_Saisie_Default
f_aux	Fraction effective du ballon chauffée par l'appoint	-	double
Valeur_Certifiee_Justifiee_Default	Choix du type de valeur pour le coefficient de perte thermique du ballon	-	RT_Valeur_Certifiee_Justifiee_Default_Ballon
Nature_Ballon	Nature du ballon pour déterminer le coefficient de perte par défaut	-	RT_Nature_Ballon
UA_S	Coefficient de pertes thermique du ballon	-	Double
Theta_Max	Température maximale du ballon	°C	double
b_sto_e	Coefficient d'atténuation dans le cas où le stockage est hors volume chauffé	-	Double
V_tot_appoint	Volume total du ballon d'appoint	L	double
Valeur_Certifiee_Justifiee_Default_Appoint	Choix du type de valeur pour le coefficient de perte thermique du ballon d'appoint	-	RT_Valeur_Certifiee_Justifiee_Default_Ballon

	oint		
Nature_Ballon_Appoint	Nature du ballon d'appoint pour déterminer le coefficient de perte par défaut	-	RT_Nature_Ballon
UA_S_appoint	Coefficient de pertes thermique du ballon d'appoint	W/K	double
Theta_Max_appoint	Température maximale du ballon d'appoint	°C	Double
type_gest_th_base	Type de gestion du thermostat du ballon de stockage d'EC S (inutile dans le cas d'une boucle solaire)	-	Int
Statut_Delta_Theta_Base	Statut_Delta_Theta_Base	-	RT_Valeur_Declaree_Default
Delta_Theta_base	Hystérésis du thermostat du ballon (inutile dans le cas d'une boucle solaire)	°C	Double
hech_base	Hauteur (relative) de l'échangeur du générateur de base à partir du fond de la cuve du ballon	-	Double
z_reg_base	Numéro de la zone du ballon qui contient le système de régulation de la base (inutile dans le cas d'une boucle solaire)	-	Int
type_gest_th_appoint	Type de gestion du thermostat du ballon de stockage d'EC S	-	RT_Gestion_Thermostat_Ballon
Statut_Delta_Theta_Appoint	Valeur_Default_Declaree_Delta_Theta_Appoint	-	RT_Valeur_Declaree_Default
Delta_Theta_appoint	Hystérésis du thermostat du ballon	°C	Double
hech_appoint	hauteur (relative) de l'échangeur du générateur d'appoint à partir du fond de la zone z_appoint	-	Double
z_appoint	Numéro de la zone	-	int

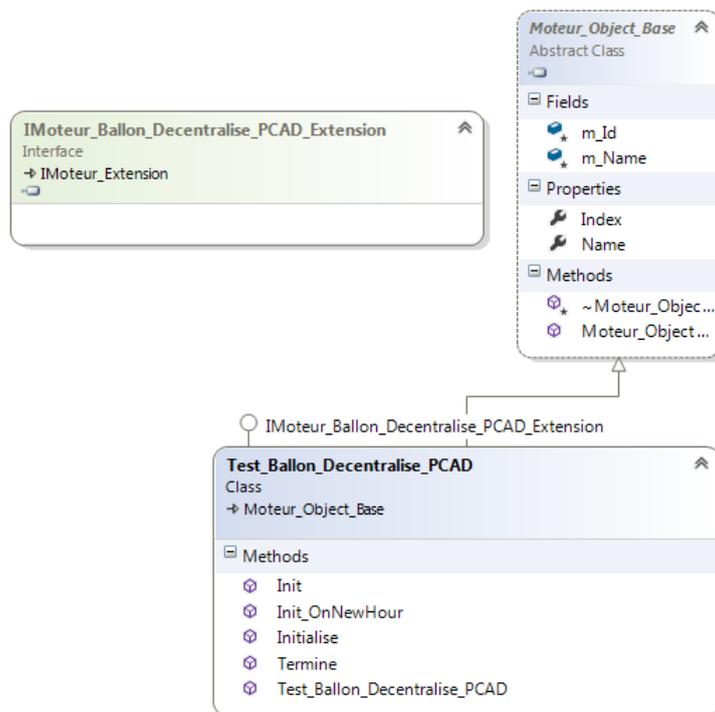
	du ballon qui contient l'élément chauffant et le thermostat de l'appoint		
z_reg_appoint	Valeur de réglage de la régulation de la boucle de chauffage	K	Int
delta_reg_bcl_ch	Valeur de réglage de la régulation de la boucle de chauffage	K	double

5.1.7.1.4 Interface *IData_Ballon_Decentralise_Extension*

Aucune propriété

5.1.7.2 Implémentation des interfaces du code de calcul

L'extension doit hériter de « *IMoteur_Ballon_Decentralise_PCAD_Extension* » et de la classe abstraite « *Moteur_Object_Base* »



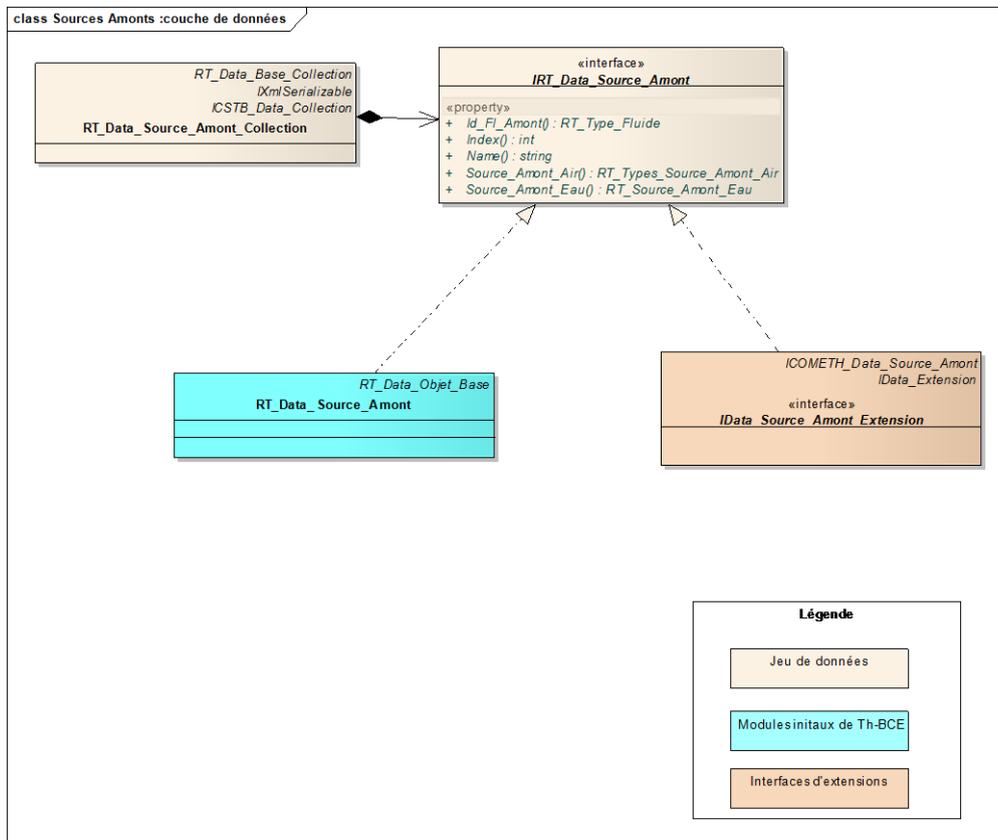
5.2 LES SOURCES AMONTS (DES GENERATEURS THERMODYNAMIQUES)

5.2.1 A PROPOS DES SOURCES AMONTS

Les sources amont couvrent la modélisation du dispositif côté source des générateurs thermodynamiques, à opposer au côté utile regroupant les émetteurs et distribution dans le bâtiment. Il s'agit donc de modéliser l'évolution de la température de fluide source à l'interface du générateur thermodynamique (ou du sol dans le cas de générateur à détente directe), ainsi que les consommations associées à la circulation ou l'adaptation de la température de ce fluide.

5.2.2 IMPLEMENTATION DANS LE JEU DE PARAMETRES

Des sources amont peuvent être rajoutées dans la collection **RT_Data_Source_Amont_Collection**.



5.2.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

Le jeu de paramètres d' entrée d' une source amont est défini par l' interface « `IData_Source_Amont` ».

La description des propriétés de l' interface « `IData_Source_Amont` » est la suivante :

Nom	Description	Unité	Type	Intervalle
Name	Nom de la source amont	-	String	-
Index	Identifiant unique d' une source amont	-	int	[0 ; +∞]
Id_Fl_Amont	Identificateur du fluide amont Eau, Air ou Sol	-	RT_Type_Fluide	
Source_Amont_Air	Type de source amont air	-	RT_Types_Source_Amont_Air	
Source_Amont_Eau	Type de source amont eau	-	RT_Types_Source_Amont_Eau	

5.2.4 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d' une source amont est défini par l' interface « IMoteur_Source_Amont ».

Entrées du composant

Nom	Description	Unité	Type	Intervalle
Generateurs_Thermodynamiques_connectes	liste des générateurs thermodynamiques connectés à la source amont		List<IMoteur_Generateur_Thermodynamique>	
VMC_DF_SF_Extraction	Référence sur un objet de ventilation mécanique de type Simple Flux Extraction ou Double Flux		IMoteur_Ventilation_Mecanique	

Sorties

Nom	Description	Unité	Type	Intervalle
Calcul_Puissance_Fournie_Maximale	Fonction « Delegate » sur le calcul de la puissance fournie maximale pour la source amont		D_Calcul_Puissance_Fournie_Maximale	
Theta_Amont_Gnr	Température amont du générateur gnr, dans le cas des générateurs thermodynamiques	°C	double	
Theta_be	Température de l'eau dans la boucle le jour j. (en cas d' une connexion à une boucle d' eau)	°C	double	
Id_Et	Identifiant de l'espace tampon sur lequel est branchée la source amont		int	
Id_DF_SF_Extraction	Identifiant de la ventilation mécanique où est localisée le simple flux		int	
Id_FI_Amont	Identificateur du fluide amont Eau, Air, Fluide		RT_Type_Fluide	
Idamont_Eau_Type	Type de source amont eau du générateur		RT_Types_Source_Amont_Eau	

Idamont_Air_Type	Type de source amont air du générateur		RT_Types_Source_Amont_Air
Pvent_Gaine	Puissances des ventilateurs dans le cas de machines sur air gainées	Wh	double
Qv_nom_tour	Débit nominal d' eau à refroidir circulant dans la tour sèche ou humide.	m3/h	double
Theta_Es_Tour_Consigne	Consigne de température d' eau en sortie de tour		double
Pvent_Tour	Puissance électrique des ventilateurs de la tour		double
Ppompes_Tour	Puissance électrique des pompes du circuit de refroidissement		double

5.3 LES ESPACES TAMPONS

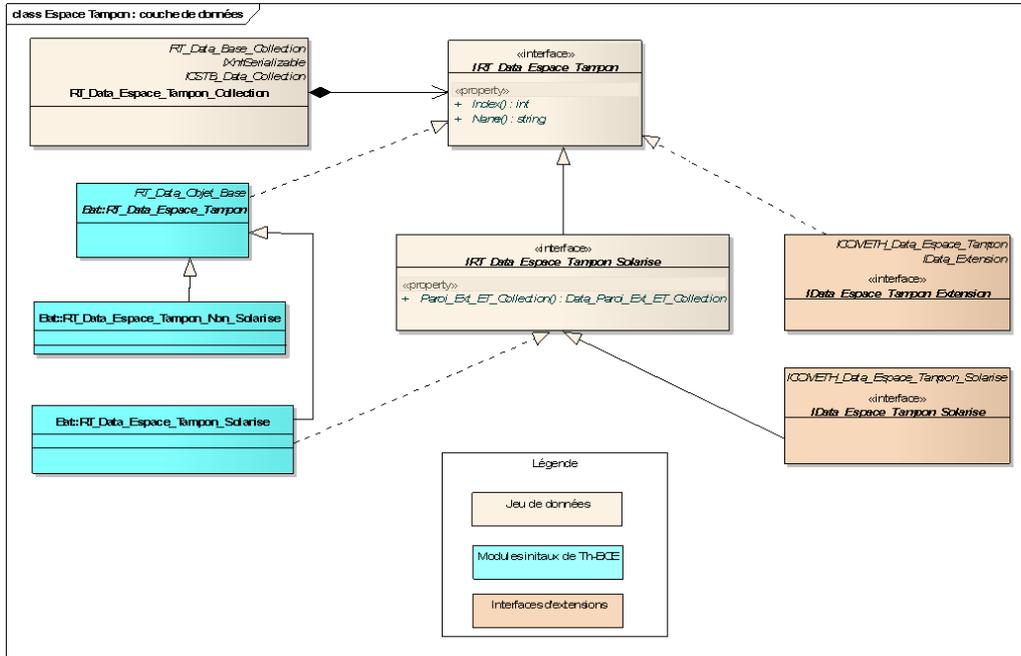
5.3.1 A PROPOS DES ESPACES TAMPONS

Les espaces tampons regroupent l'ensemble des espaces séparant l'environnement extérieur de l'environnement intérieur. En particulier, ils permettent la prise en compte des locaux non-chauffés (ou chauffés à une température inférieure à 12°C en occupation).

5.3.2 IMPLEMENTATION DANS LE JEU DES PARAMETRES

Remarque : une bonne connaissance de la structure des données du moteur de calcul est nécessaire pour la suite.

Des espaces tampons peuvent être rajoutées dans la collection RT_Data_Espace_Tampon_Collection.



5.3.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

	Non Solarise	Solarise
IData_Espace_Tampon_Extension	X	X
IData_Espace_Tampon_Solarise		X

Le jeu de paramètres d' entrée d' un espace tampon dépend de la nature de ce dernier. Il peut être soit solarisé, soit non solarisé comme l' indique le tableau ci-dessus.

La description des propriétés de l' interface « IData_Espace_Tampon_Extension » est la suivante :

Nom	Description	Unité	Type	Intervalle
Name	Nom de l' espace tampon	-	String	-
Index	Identifiant unique d' un espace tampon.	-	int	[0 ; +∞]

Si l' espace tampon est solarisé, l' interface « `IData_Espace_Tampon_Solarise` » doit aussi être implémentée. La description de ces propriétés est la suivante :

Nom	Description	Type
<code>Paroi_Ext_ET_Collection</code>	Collection des parois des espaces tampons	<code>Data_Paroi_Ext_ET_Collection*</code>
<code>Paroi_Ext_Espace_Tampon_Collection</code>	Collection des composants d'entrée internes au moteur des parois extérieures d'espaces tampons.	<code>List<ICOMETH_Data_Paroi_Ext_Espace_Tampon>**</code>

(*) La collection `Data_Paroi_Ext_ET_Collection` est une collection de jeu donnée d'entrée des parois qui séparent l'espace tampon de l'extérieur, telles qu'elles sont conçues dans la RT2012. Cette collection a été rendu disponible aux développeurs d'extension dynamique de type « espace tampon ». Elle est située dans l'espace de noms « `RT2012.Extensions.EspaceTampon` ».

(**) Le paramètre `Paroi_Ext_Espace_Tampon_Collection` est une liste de composants d'entrée des parois qui séparent l' espace tampon de l' extérieur, internes au moteur de calcul de la RT2012 appelé « `ICOMETH_Data_Paroi_Ext_Espace_Tampon` ». Cette liste est automatiquement remplie par le moteur sur la base du paramètre d' entrée précédent (`Data_Paroi_Ext_ET_Collection`) par un préprocesseur interne au moteur. Elle n' est donc pas remplie par l' utilisateur. Par conséquent, il faut lors de son implémentation, ajouter le décorateur [`XmlIgnore`] pour qu' elle ne soit pas prise en compte par le Sérialiseur Xml.

Pour en savoir plus sur ces parois, se référer au paragraphe 7.5, page 132 de la méthode Th-BCE 2012.

5.3.4 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d' un espace tampon est défini par l' interface « `IMoteur_Espace_Tampon_Extension` » et « `IMoteur_Espace_Tampon_Solarise` » si ce dernier est solarisé.

	Non Solarise	Solarise
<code>IMoteur_Espace_Tampon_Extension</code>	X	X
<code>IMoteur_Espace_Tampon_Solarise</code>		X

La description des propriétés de l' interface « IMoteur_Espace_Tampon_Extension » est la suivante :

Sorties du composant

Nom	Description	Unité	Type	Intervalle
Index	Identifiant unique d' un espace tampon	-	int	[0 ; +∞]
Theta	Température d' air dans l' espace tampon au pas de temps h.	°C	double	[-273 ; +∞]
b_therm	Coefficient de réduction des déperditions thermiques dû à la présence de l' espace tampon	-	double	[-∞ ; 1]
b_solaire	Coefficient de réduction des flux solaires dû à la présence de l' espace tampon	-	double	[0 ; 1]
b_ecl	Coefficient de réduction de l' éclairage dû à la présence de l' espace tampon (concerne uniquement les baies vitrées)	-	double	[0 ; 1]

L' interface contient aussi deux propriétés particulières. Ces dernières sont instanciés et modifiés par le moteur sur la base des sorties de l' espace tampon. Il s' agit d' une liste de composants enveloppe (baie, paroi opaque, linéaire..) et système (distributions, générations) connectés à l' espace tampon.

Nom	Description	Type
Elements_Enveloppe_Connectes_Collection	Collection des éléments enveloppe connectés à l'espace	IList <IMoteur_Element_Enveloppe_Connecte_EspaceTampon>

	tampon	
Elements_Systeme_Connectes_Collection	Collection des éléments système connectés à l'espace tampon	IList <IMoteur_Element_Systeme_Connecte_Espace_Tampon>

Le développeur d' extension dynamique doit donc uniquement implémenter cette propriété de l' interface en créant un « getteur » et un setteur », le moteur se chargeant de les remplir.

L' interface « IMoteur_Espace_Tampon_Extension » déclare également les méthodes « Init » et « Run » respectivement, appelées en début de calcul et à chaque pas de temps.

Pour les espaces tampon solarisés, on a accès aux composants Moteur des parois qui séparent l' espace tampon de l' extérieur.

La description des propriétés de l' interface « IMoteur_Espace_Tampon_Solarise » est la suivante :

Nom	Description	Type
Paroi_Ext_Espace_Tampon_Collection	Collection des composants Moteur des parois qui séparent l' espace tampon de l' extérieur	IMoteur_Paroi_Ext_Espace_Tampon_Collection : IList<IMoteur_Paroi_Ext_Espace_Tampon>

L' interface « IMoteur_Paroi_Ext_Espace_Tampon_Collection » joue deux rôles. D' une part, il s' agit d' une liste d' interfaces de composants moteur des parois de l' espace tampon (IMoteur_Paroi_Ext_Espace_Tampon), par héritage. D' autre part, elle contient une donnée en sortie :

Nom	Description	Unité	Type	Intervalle
Aue_Total	Surface total des parois de l' espace tampon donnant sur l' extérieur	M ²	double	[0; +∞]

Pour rappel, cette liste de composants « IMoteur_Paroi_Ext_Espace_Tampon » est remplie par le moteur. C' est le moteur qui s' occupe d' en appeler les méthodes. Le développeur d' extension dynamique a accès aux données suivantes :

Données du composant

Nom	Description	Unité	Type	Intervalle
FfDir	Facteur d' affaiblissement du rayonnement solaire direct du au masque de la paroi (m)	-	doubl e	[0 , 1]
FfDir	Facteur d' affaiblissement du rayonnement solaire diffus dû au masque de la paroi (m)	-	doubl e	[0 , 1]
Index	Identifiant unique d' une paroi d' espace tampon	-	int	[0 ; +∞]
Id_Mas	Indices des masques attachés à la paroi.	-	int[]	[[0 ; +∞]]
Id_Ori	Indice de l' orientation dans le tableau des orientations (α, β).	-	int	[0 ; +∞]
Rouv_max	Vecteur des ratios d'ouverture maximale des N parois de l'espace tampon	-	double	[0 ; 1]
Swe_ap	Vecteur des facteurs solaires globaux des N parois de l'espace tampon donnant sur l'extérieur, avec protection mobile en place	-	double	[0 ; 1]
Swe_sp	Vecteur des facteurs solaires globaux des N parois de l'espace tampon donnant sur l'extérieur, sans protection mobile en place	-	double	[0 ; 1]
Tle_ap	Vecteur des facteurs de transmission lumineux globaux des N parois de l'espace tampon donnant sur l'extérieur, avec protection mobile en place	-	double	[0 ; 1]
Tle_sp	Vecteur des facteurs de transmission lumineux globaux des N parois de l'espace	-	double	[0 ; 1]

	tampon donnant sur l'extérieur, sans protection mobile en place			
Uue	Vecteur des coefficients de transmission thermique des N parois de l'espace tampon donnant sur l'extérieur	-	double	[0 ; 1]
Aue	Surfaces de la paroi de l' espace tampon donnant sur l' extérieur	m ²	double	[0 ; +∞]
Isr	Vecteur des rayonnements solaires globaux atteignant les différentes parois extérieures de l' espace tampon.	W/m ²	double	[0 ; +∞]
Etp	Eclairement incident total sur la paroi	-	double	[0 ; +∞]

5.4 LA DISTRIBUTION INTERGROUPE

5.4.1 A PROPOS DE LA DISTRIBUTION INTERGROUPE

Dans le calcul règlementaire, les réseaux de distribution de chauffage, de refroidissement et d'ECS sont décomposés en deux niveaux: un propre au groupe (intérieurs des locaux desservis), l'autre intergroupes (partie collective du réseau, colonnes, réseau entre bâtiments

Les réseaux de distribution intergroupes font donc communiquer les besoins en énergie et températures d'eau des différents groupes avec la génération d'énergie (chaufferie, générateur...).

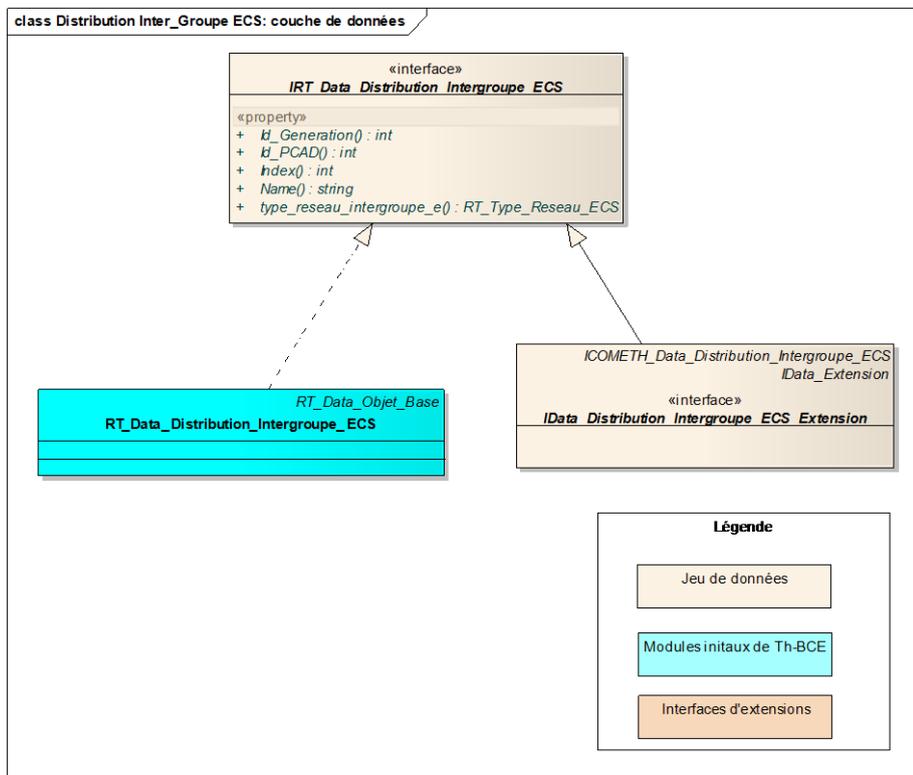
Ils intègrent les calculs des pertes de distribution et de consommation de circulateur de la partie collective du réseau de distribution.

Les réseaux de distribution intergroupes font donc communiquer les besoins en énergie et températures d'eau des différents groupes avec la génération d'énergie (chaufferie, générateur...). Ils intègrent les calculs des pertes de distribution et consommation de circulateur de la partie collective du réseau de distribution.

Dans le cadre de la méthode règlementaire, les réseaux de distribution intergroupes sont compartimentés entre le chauffage, le refroidissement et l'ECS. L'interface d'extension dynamique permet de remplacer les composants distribution intergroupes de chauffage ou d'ECS, mais également développer une famille de composants mixte chauffage et ECS inédite par rapport à la méthode Th-BCE.

5.4.2 La distribution intergroupe ECS

5.4.2.1 IMPLEMENTATION DANS LE JEU DES PARAMETRES



La distribution intergroupe ECS peut être rajoutée dans la collection `Distribution_Intergroupe_ECS_Collection` au niveau de `RT_Data_Projet`.

La description des propriétés de l'interface « `IData_Distribution_Intergroupe_ECS_Extension` » est la suivante :

Nom	Description	Unité	Type	Intervalle
Index	Indice unique de la distribution	-	Int	[0; +∞]
Name	Nom de la distribution	-	String	
Id_Generation	Indice de la génération	-	Int	[0; +∞]

Id_PCAD	Identifiant du PCAD associé à la partie ECS de la distribution	-	Int	[0; +∞]
Type_reseau_inetrgroupe_ecs	Type du réseau de distribution intergroupe ECS (0 : Pas de réseau, 1 : Bouclé, 2 : Tracé)	-	RT_Type_Reseau_ECS	[0 ; 2]

La propriété `Dist_Second_ECS_Collection` est instanciée et remplie par le moteur de calcul.

5.4.2.2 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d' une distribution intergroupe ECS est défini par l' interface « `IMoteur_Distribution_Intergroupe_ECS_Extension` »

La description de l' interface « `IMoteur_Distribution_Intergroupe_ECS_Extension` » est la suivante :

Entrées du composant

Nom	Description	Unité	Type	Intervalle
Adess_e	Surface totale des émetteurs desservis par cette distribution primaire. Cette entrée est remplie par le moteur.	-	Int	[0; +∞]
B2nd	Coefficient de réduction des déperditions associé à un espace tampon (Cette entrée n' est pas utilisée dans ce cadre)	-	double	
Dist_Second_ECS_Collection	Collection des distributions secondaire ECS du groupe	-	IMoteur_Distribution_Group_e_ECS_Collection	

List_Gr	Liste des index des groupes liés à la génération via des réseaux de chaud. Cette entrée est remplie par le moteur.	-	List<int>	-
Rat_Bes_Gr_Ecs	Ratio de la surface d'un émetteur ECS équivalent relié à ce réseau primaire d'ECS sur la somme des surfaces de tous les émetteurs ECS équivalents reliés à ce réseau primaire d'ECS. Cette entrée est remplie par le moteur.	-	Dictionnaire<int, double>	[0,1]
Rat_Surf_Gr_Ecs	Ratio de la surface d'un émetteur ECS équivalent relié à ce réseau primaire d'ECS sur la somme des surfaces de tous les émetteurs ECS équivalents reliés à ce réseau primaire d'ECS. Cette entrée est remplie par le moteur.	-	Dictionnaire<int, double>	[0,1]
Sum_Agr_e	Surface totale des groupes desservie par le réseau de distribution primaire. Cette entrée est remplie par le moteur.	m ²	double	[0; +∞]

Sortie du composant

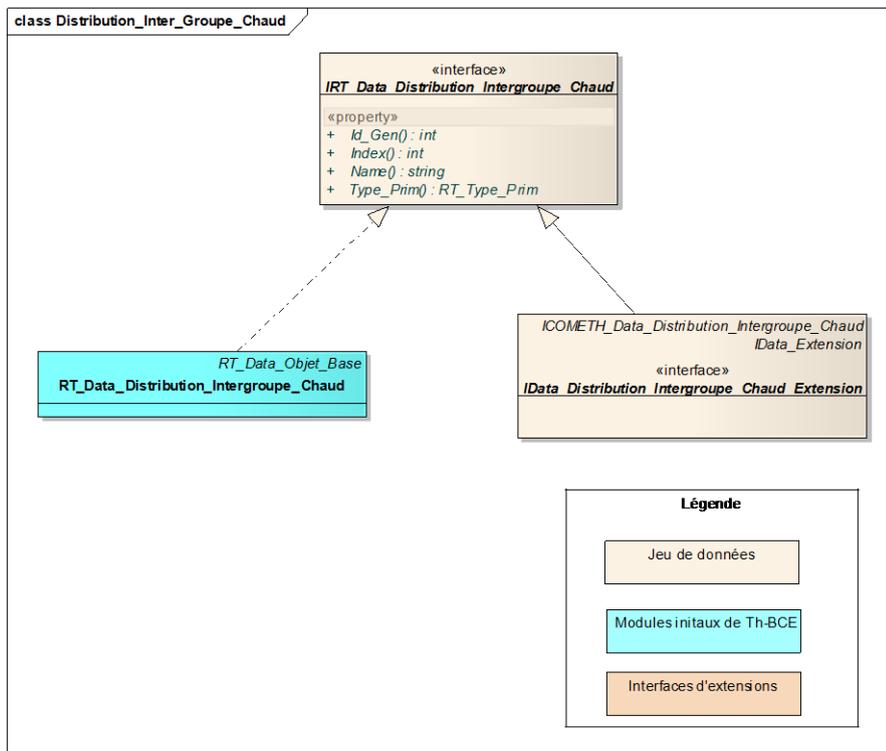
Nom	Description	Unité	Type	Intervalle
id_enc1_e	Indicateur qui précise si le jour courant est inclut dans la période de fonctionnement de la génération	-	int	[0;1]
Id_Et	Identifiant de l'espace tampon associé. Cette sortie n' est pas utilisée par le moteur	-	Int	[0; +∞]
Index	Index unique de la distribution ; Idem à l' index de l' objet Data	-	Int	[0; +∞]

phi_pertes_hvc_prim_e	Pertes thermiques horaires hors volume chauffé du réseau de distribution primaire d'ECS	Wh	double	[0; +∞]
phi_pertes_vc_prim_e	Pertes thermiques horaires en volume chauffé du réseau de distribution primaire d'ECS	Wh	double	[0; +∞]
Phi_Sys_Prim_Ecs	Besoins ECS majorés des pertes de distribution secondaire et primaire	Wh	double	[0; +∞]
theta_moy_prim_e	Température moyenne du réseau de distribution primaire d'ECS	°C	double	[0; +∞]
W_aux_e	Consommation électrique des auxillaires de distribution primaire d'ECS en énergie finale	Wh	double	[0; +∞]
W_rechauf_e	Consommation électrique de l'éventuel réchauffeur de la distribution primaire bouclée d'ECS, en énergie finale	Wh	double	[0; +∞]

.

5.4.3 La distribution intergroupe chaud

5.4.3.1 IMPLEMENTATION DANS LE JEU DES PARAMETRES



La distribution intergroupe chaude peut être rajoutée dans la collection `Distribution_Intergroupe_Chaud_Collection` au niveau de `RT_Data_Projet`.

La description des propriétés de l'interface « `IData_Distribution_Intergroupe_Chaud_Extension` » est la suivante :

Nom	Description	Unité	Type	Intervalle
Index	Indice unique de la distribution	-	Int	[0; +∞]
Name	Nom de la distribution	-	String	
Id_Genera	Indice de la génération	-	Int	[0; +∞]

tion

Type_Prim _Chaud	Configuration du réseau (0 : Réseau de distribution fictif,1 : Réseau de distribution hydraulique,2 : Raccordement hydraulique direct à la génération (sans pertes))	-	RT_Typ e_Prim	[0 ;2]
---------------------	--	---	------------------	--------

Remarque :

La propriété `Ges_Reg_Primaire_Chaud` est instanciée et remplie par le moteur de calcul.

5.4.3.2 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d' une distribution intergroupe ECS est défini par l' interface « `IMoteur_Distribution_Intergroupe_Chaud_Extension` »

La description de l' interface « `IMoteur_Distribution_Intergroupe_Chaud_Extension` » est la suivante :

Entrées du composant

Nom	Description	Unité	Type	Intervalle
<code>Ges_Reg_Primaire_Chaud</code>	Un accesseur à la gestion régulation de la distribution intergroupe chaud. Cette propriété est à implémenter	-	<code>IMoteur_Gestion_Regulation_Intergroupe_Chaud</code>	
<code>List_Gr</code>	Liste des index des groupes liés à la génération via des réseaux de chaud. Cette entrée est remplie par le moteur.	-	<code>List<int></code>	-

Sortie du composant					
Nom	Description	Unité	Type	Intervalle	
Id_Et	Identifiant de l'espace tampon associé. Cette sortie n' est pas utilisée par le moteur	-	Int	[0; +∞]	
Index	Index unique de la distribution, Idem à l' index de l' objet Data	-	Int	[0; +∞]	
Phi_Aux_Vc	Consommations d' énergie des circulateurs transmise au volume chauffé sous forme de chaleur au pas de temps h	Wh	double	[0; +∞]	
Phi_Pertes_Vc	Pertes thermiques du réseau de distribution primaire en volume chauffé	Wh	double	[0; +∞]	
theta_moy_prim_e	Température moyenne dans le réseau primaire, qui va permettre de déterminer le comportement du générateur	°C	double	[0; +∞]	
Phi_Sys_Prim	Besoin en énergie de chauffage/refroidissement augmenté des pertes thermiques du réseau primaire	Wh	double	[0; +∞]	
W_aux	Energie consommée par le circulateur du réseau de distribution primaire au pas de temps h	Wh	double	[0; +∞]	

Le code de calcul de la gestion régulation primaire chaud est défini par Ges_Reg_Primaire_Chaud est défini par l'interface « IMoteur_Gestion_Regulation_Intergroupe_Chaud ».

La description de l' interface « IMoteur_Gestion_Regulation_Intergroupe_Chaud » est la suivante :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Sum_Agr_Prim	Surface totale des groupes desservie par le réseau de distribution primaire	m ²	double	[0; +∞]
Adess	Ratio de la surface desservie dans le groupe gr sur la surface totale desservie par la distribution primaire, en chauffage.	-	double	[0; 1]
Dist_Second_Chaud_Collection	Collection des distributions secondaire Chaud du groupe	-	IMoteur_Distribution_Groupe_Chaud_Collection	
Rat_Bes_Gr_Ch	Ratio des besoins de chauffage - refroidissement d' un groupe relié à un réseau primaire sur la somme de tous les besoins de chauffage/refroidissement du réseau primaire	-	Dictionnaire<int, double>	[0,1]
Rat_Surf_Gr_Ch	Ratio de la surface du groupe gr sur la surface totale desservie par le réseau de distribution primaire en chauffage/refroidissement	-	Dictionnaire<int, double>	[0,1]
Rat_Surf_Dess_Gr	Ratio de la surface desservie dans le groupe gr sur la surface totale desservie par la distribution primaire, en chauffage/refroidissement	-	Dictionnaire<int, double>	[0,1]

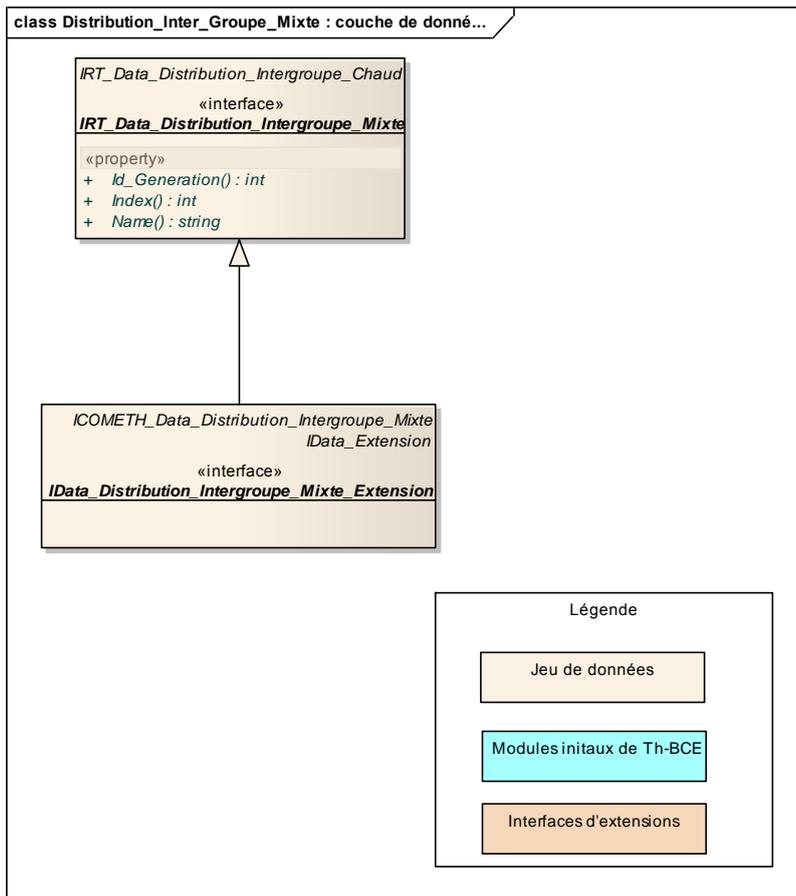
Phi_ds_req	Energie de chauffage/refroidissement totale requise en entrée du réseau primaire	W	double	[0; +∞]
Qresid	Débit résiduel minimal dû à des circulateurs en régime de fonctionnement permanent	m3/h	double	[0; +∞]
Qtot_Req	Débit total requis par les réseaux de distributions secondaires	m3/h	double	[0; +∞]
Qnom	Débit nominal du réseau primaire (dimensionnement)	m3/h	double	[0; +∞]
Theta_Depart	Température du départ du réseau primaire	°C	double	[-50; +∞]
Mod_pertes	Coefficient de modulation des pertes de réseaux en fonction de leur gestion	-	double	[0; 1]
Delta_Fonc	Indicateur de débit dans le réseau primaire	-	Int	{0,1}
Theta_Max_Ch	Température maximale de la distribution intergroupe en chauffage	°C	double	[-50; +∞]

Sortie du composant

Nom	Description	Unité	Type	
Theta_air_aval_eq	Température d' air équivalente vu par la distribution intergroupe.	°C	double	[-50; +∞]
Type_Prim_Chaud	Type de réseau de distribution primaire 0 : réseau de distribution virtuel sans pertes, 1 : réseau de distribution physique	RT_Type_2nd		{0 ;1}

5.4.4 La distribution intergroupe mixte

5.4.4.1 IMPLEMENTATION DANS LE JEU DES PARAMETRES



La distribution intergroupe mixte peut être rajoutée dans la collection **Distribution_Intergroupe_Mixte_Collection** au niveau de **RT_Data_Projet**.

La description des propriétés de l'interface « IData_Distribution_Intergroupe_Mixte_Extension » est la suivante :

Nom	Description	Unité	Type	Intervalle
Index	Indice unique de la distribution	-	Int	[0; +∞]
Name	Nom de la distribution	-	String	
Id_Generation	Indice de la génération	-	Int	[0; +∞]
Type_Prim_Chaud	Configuration du réseau (0 : Réseau de distribution fictif, 1 : Réseau de distribution hydraulique, 2 : Raccordement hydraulique direct à la génération (sans pertes))	-	RT_Type_Prim	[0 ; 2]
Id_PCAD	Identifiant du PCAD associé à la partie ECS de la distribution	-	Int	[0; +∞]
Type_reseau_inetrgroupe_ecs	Type du réseau de distribution intergroupe ECS (0 : Pas de réseau, 1 : Bouclé, 2 : Tracé)	-	RT_Type_Reseau_ECS	[0 ; 2]

5.4.4.2 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d' une distribution intergroupe ECS est défini par l' interface « `IMoteur_Distribution_Intergroupe_Mixte_Extension` »

La description de l' interface « `IMoteur_Distribution_Intergroupe_Mixte_Extension` » est la suivante :

Nom	Description	Unité	Type	Intervalle
<code>Distrib_Intergp_Chaud</code>	Distribution intergroupe chaud	-	<code>IMoteur_Distribution_Intergroupe_Chaud</code>	[1;1]
<code>Distrib_Intergp_ECS</code>	Distribution intergroupe ECS	-	<code>IMoteur_Distribution_Intergroupe_ECS</code>	[1;1]

Remarque :

Il faut instancier les 2 distributions ECS et chaud dans la constructeur de la classe implémentant l' interface `IMoteur_Distribution_Intergroupe_Mixte_Extension`.

Si vous voulez faire communiquer les deux distributions intergroupes (ECS & Chaud) entre elles, vous pouvez référencer une dans le constructeur de l' autre.

L' objet Data de l' élément moteur `IMoteur_Distribution_Intergroupe_Mixte_Extension` est de type `ICOMETH_Data_Distribution_Intergroupe_Mixte`, ce dernier est casté par le moteur à l' utilisation des datas distribution intergroupe ECS et mixte.

5.5 LES PRODUCTIONS STOCKAGE

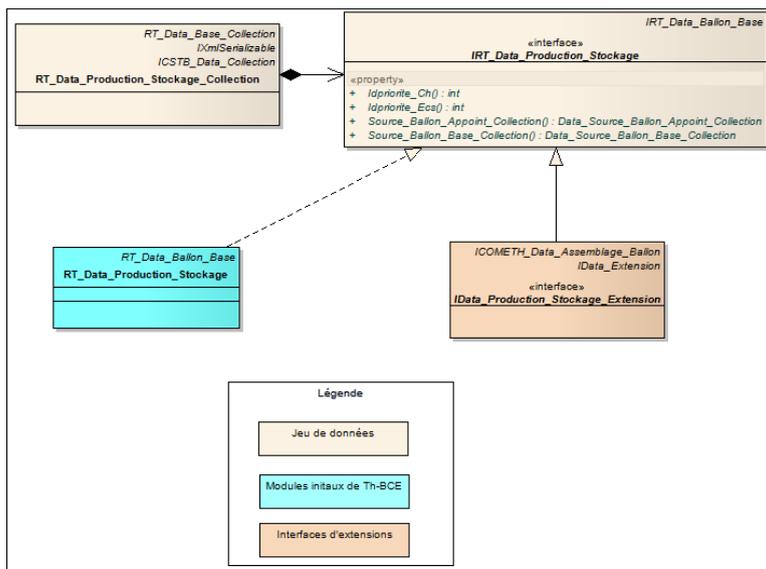
5.5.1 A PROPOS DES PRODUCTIONS STOCKAGE

L' objet production stockage est utilisé pour la modélisation des ballons de stockage fournissant de l' eau chaude dédiée à l' ECS ou au chauffage. Un ou deux générateurs sont associés au ballon (le générateur de base et, le cas échéant, le générateur d' appoint) et sont utilisés pour le chauffer à sa température de consigne. A chaque pas de temps, le moteur envoie à la production stockage l' énergie requise en ECS et/ou en chaud calculée par la génération. La production stockage répondra à cette demande selon ses capacités en retournant la puissance qu' elle a pu fournir. Le puisage d' énergie dans la production stockage aura pour conséquence de la refroidir, mais ce refroidissement pourra être compensé par l' appel de puissance du (des) générateur(s).

5.5.2 IMPLEMENTATION DANS LE JEU DE DONNEES

Remarque : une bonne connaissance de la structure des données du moteur de calcul est nécessaire pour la suite.

Des productions stockage peuvent être rajoutées dans la collection `RT_Data_Production_Stockage_Collection`.



5.5.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

Le jeu de paramètres d' entrée d' un composant production stockage est défini par l' interface : « `IData_Production_Stockage_Extension` ».

Les paramètres remplis par l' utilisateur associés à cette interface sont :

Nom	Description	Unité	Type	Intervalle
Name	Nom du composant	-	String	-
Index	Identifiant unique d'un composant	-	int	[0 ; +∞]
Rdim	Nombre de composants identiques	-	int	[1 ; +∞]
Id_Fl_Aval	Identificateur du fluide aval Eau(1) Air(2) Sol(3)	-	E_Id_Fluide	[1 ; 3]
Id_Fou_Gen	Fonction du composant en tant que générateur (Ch, Fr, Ecs, Ch+Ecs ou Ch+Fr)	-	RT_Types_ Fonctionnements_Generateur	
Idpriorite_Ch	Indice de priorité du générateur en chauffage	-	int	[1 ; +∞]
Idpriorite_Ecs	Indice de priorité du générateur en ECS	-	int	[1 ; +∞]
Id_Fou_Sto	Fonction du composant en tant qu'assemblage ballon (Ch, Ecs ou Ch+Ecs)	-	RT_Id_FouGen_Mode1	-
Data_Source_Ballon_Base_Collection	Générateurs de base du ballon		Data_Source_Ballon_Base_Collection **	
Data_Source_Ballon_Appoint_Collection	Générateurs d' appoint du ballon		Data_Source_Ballon_Appoint_Collection **	

(**) Les collections *Data_Source_Ballon_Base/Appoint_Collection* sont des collections de jeu de données d' entrée des generateurs de base/appoint du ballon, telles qu' elles sont conçues dans la RT2012. Cette collection a été rendu disponible aux développeurs d' extension dynamique de type « production stockage ». Elle est située dans l' espace de noms « *RT2012.Extensions.Production_Stockage* ».

L' interface comporte aussi la propriété suivante :

Nom	Description	Type
-----	-------------	------

Assemblage_Echangeur_Generateur_Ballon_	Référence vers les sources de chaleur	List <ICOMETH_Data_Assemblage_Echangeur_Generateur_Ballon>***
--	---------------------------------------	---

(***) Le paramètre *Assemblage_Echangeur_Generateur_Ballon_Collection* est une liste de composants d'entrée des assemblages générateurs reliés au ballon, internes au moteur de calcul de la RT2012 appelés « *ICOMETH_Data_Assemblage_Echangeur_Generateur_Ballon* ». Cette liste est automatiquement remplie par le moteur sur la base des paramètres d'entrée précédents (*Data_Source_Ballon_Base/Appoint_Collection*) par un préprocesseur interne au moteur. Elle n'est donc pas remplie par l'utilisateur. Par conséquent, il faut lors de son implémentation, ajouter le décorateur [*XmlIgnore*] pour qu'elle ne soit pas prise en compte par le Sérialiseur Xml.

5.5.4 L' INTERFACE DU CODE DE CALCUL

La partie algorithme de la production stockage extension dynamique hérite toujours de l'interface *IMoteur_Production_Stockage_Extension*. Son contenu en variable est le suivant :

Entrées du composant

Nom	Description	Unité	Type	Intervalle
Theta_amb	Température d'ambiance du lieu où se trouve le générateur	°C	double	[-∞ ; +∞]
IdEcs_Seul	Indication de production ECS seule	-	int	[1 ; +∞]
Generateur_Ballon	référence vers le tableau des générateurs branchés au ballon	List< <i>IMoteur_Assemblage_Generateur_Echangeur_Pour_Ballon</i> >		

Sorties

Nom	Description	Unité	RT_Types_Fonctionnements_Generateur	
Id_Fou_Gen	Fonctionnement du générateur	-		-
Qrest	Energie restant à fournir à la fin du pas de temps, faisant l'objet d'un report de demande à un autre générateur en séquence ou au pas de temps suivant	Wh	double	[0 ; +∞]
Phi_vc	Pertes thermiques et puissances des auxiliaires du générateur transmises vers l'ambiance	Wh	double	[0 ; +∞]
Qprelec	Production électrique du générateur	Wh	double	[0 ; +∞]
Waux_pro	Consommation des auxiliaires au pas de temps h	Wh	double	[0 ; +∞]

Pmax	Puissance maximale du générateur	W	double	[0 ; +∞]
Q_cons	Puissance consommée par le générateur	Wh	double	[0 ; +∞]
Taux_Charge	Taux de charge du générateur	-	double	[0 ; 1]
Q_fou	Puissance fournie par le générateur	W	double	[0 ; +∞]
Pn_Gen_Ch	Puissance nominale du générateur en chauffage	W	double	[0 ; +∞]
Q_Cef_Ch	Tableau des consommations en énergie finale pour le chauffage en fonction des énergies utilisées (Gaz, Fioul, Charbon, Bois, Electricité, Réseau de fourniture)	Wh	RT_Consommations_Energies	[0 ; +∞]
R_fonctecs	Temps de fonctionnement du générateur pour la production d'ECS, à puissance maximale, au pas de temps h	-	double	[0 ; 1]
Pn_Gen_Ecs	Puissance nominale du générateur en ECS	W	double	[0 ; +∞]
Q_Cef_Ecs	Tableau des consommations en énergie finale pour le refroidissement en fonction des énergies utilisées (Gaz, Fioul, Charbon, Bois, Electricité, Réseau de fourniture)	Wh	RT_Consommations_Energies	[0 ; +∞]

Cette interface déclare deux méthodes nommée « Run_Ch » pour le chauffage, et « Run_Ecs » pour l'ECS. Chacune de ces méthodes prend les paramètres suivant :

Nom	Description	Unité	Type	Intervalle
aQreq	Energie requise en entrée du générateur	Wh	double	[0 ; +∞]
aThetaAval	Température avale	°C	double	[-∞ ; +∞]
aRat_puiDispo	Ratio de puissance disponible	-	double	[0 ; 1]

Ces méthodes retournent la puissance fournie par le générateur :

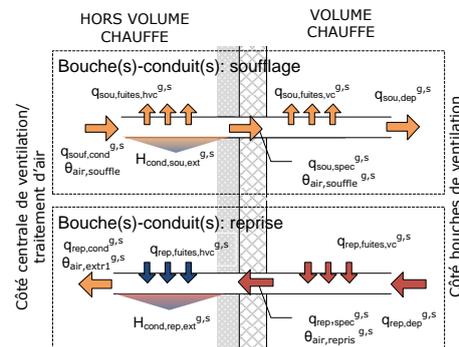
Nom	Description	Unité	Type	Intervalle
-	Energie totale effectivement fournie par le générateur au pas de temps h	Wh	double	[0 ; +∞]

5.6 LES VENTILATION MECANIQUES

5.6.1 A PROPOS DES VENTILATIONS MECANIQUES

Le composant ventilation mécanique représente la centrale de ventilation ou de traitement d' air. Il permet de modéliser la régulation centrale des débits d' air, des températures et humidités d' air soufflés, ainsi que des consommations de ventilateurs. Les conduits et bouches complétant le dispositif de ventilation/traitement d' air sont prises en compte par ailleurs au travers de composants bouche(s)-conduit(s) avec lequel le composant ventilation mécanique communique.

Les conventions de notations de la méthode Th-BCE relatives aux composants bouches-conduits sont les suivantes :



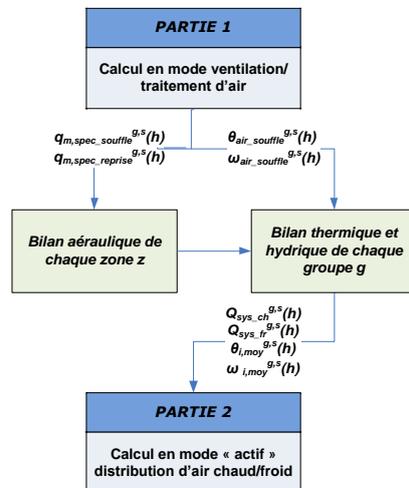
L' ensemble de ces données sont calculées selon les hypothèses et le paramétrage des composants bouches-conduits (méthode Th-BCE chapitre 8.2). Il est possible de récupérer et modifier ces valeurs dans le composant ventilation mécanique.

A chaque pas de temps horaire de simulation, le calcul relatif à la ventilation mécanique et traitement d' air est séquencé en 2 étapes :

- **Partie 1** : calcul des débits et caractéristiques de l' air soufflé. Le composant de ventilation/traitement d' air est modélisé selon l' hypothèse que les besoins de chauffage et refroidissement sont nuls (modes ventilation/traitement d' air seul). Les fonctions de préchauffage ou pré refroidissement non basées sur les besoins de chauffage/refroidissement sont également prises en compte à ce stade. Ce calcul permet d' alimenter les bilans aérauliques (équilibrage des débits par zone), puis thermique (calcul des températures et besoins de chauffage/refroidissement par groupe). Exemples d' algorithmes « Partie 1 » : chapitres 8.4 à 8.6 de la méthode Th-BCE. Chapitre « 8.3 C_Ven_CTA DAV » jusqu' au paragraphe 8.3.3.4 inclus.

- **Partie 2** : cette seconde partie de calcul concerne les systèmes possédant des fonctions d' émission/distribution de chauffage/refroidissement des locaux (dépendant de la charge thermique de ceux-ci) en plus du rôle de ventilation. Une fois les besoins de chauffage/refroidissement calculés au niveau de chaque groupe selon les hypothèses de la partie 1, on reprend la modélisation de la centrale, en prenant en compte l' influence de la charge thermique sur son point de fonctionnement. Les besoins de chauffage et refroidissement sont éventuellement corrigés pour intégrer l' impact du basculement en mode actif (notamment en cas d' augmentation des débits).

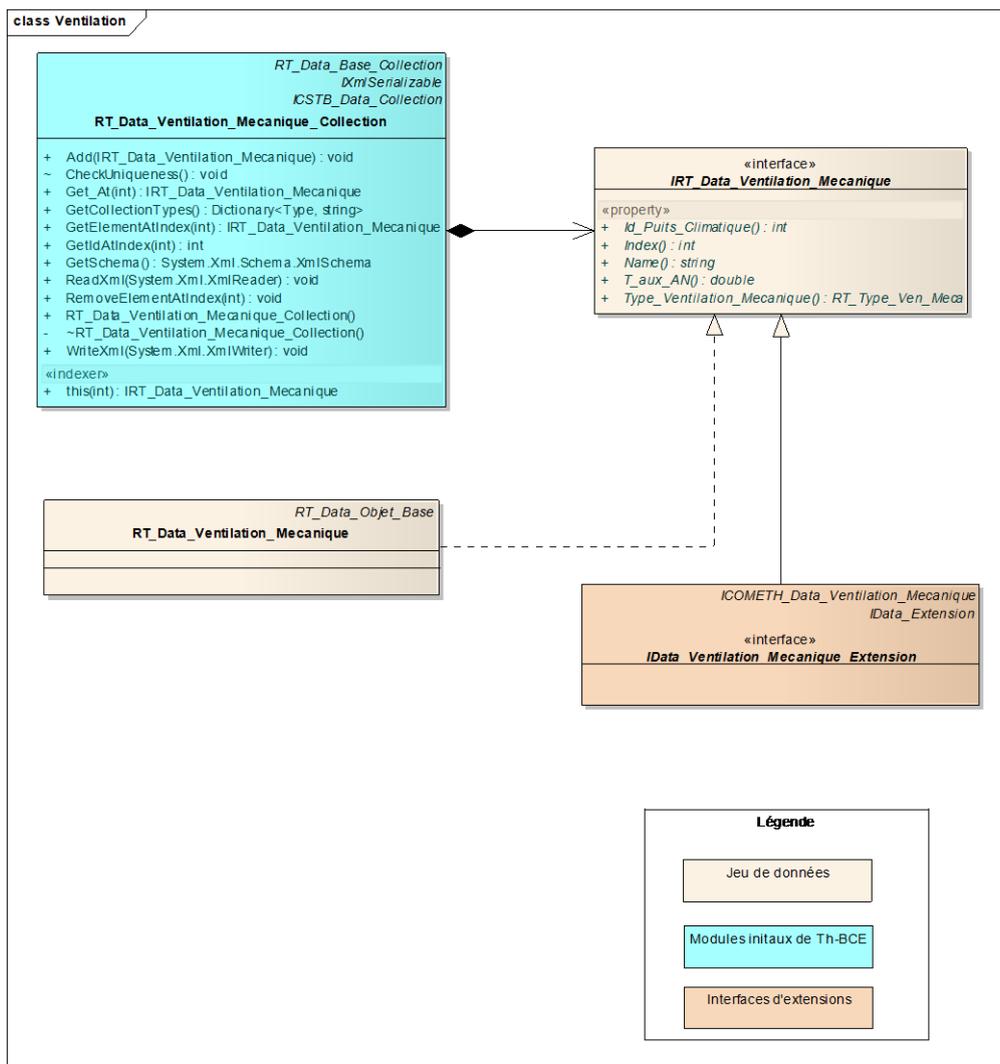
Exemple d' algorithme « Partie 2 » : chapitre « 8.3 C_Ven_CTA DAV » à partir du paragraphe 8.3.3.5. La CTA DAV est le seul composant de la méthode Th-BCE a exploité le calcul partie 2.



5.6.2 IMPLEMENTATION DANS LE JEU DE DONNEES

Remarque : une bonne connaissance de la structure des données du moteur de calcul est nécessaire pour la suite.

Des ventilations mécaniques peuvent être rajoutées dans la collection `RT_Data_Ventilation_Mecanique_Collection`.



5.6.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

	Non DAV	DAV
IData_Ventilation_Mecanique_Extension	X	X
IData_Ventilation_Mecanique_DAV		X

Le jeu de paramètres d' entrée d' une ventilation mécanique dépend de la nature de cette dernière. Elle peut être soit à débit variable, soit pas comme l' indique le tableau ci-dessus.

La description des propriétés de l' interface « IData_Ventilation_Extension » est la suivante :

Nom	Description	Unité	Type	Intervalle
Name	Nom du composant	-	String	-
Index	Identifiant unique d'un composant	-	int	[0 ; +∞]
Type_Ventilation_Mecanique	Type du système mécanique de ventilation	-	RT_Type_Ven_Meca	[0 ; 7]
T_aux_AN	Taux d'air neuf	-	Double	[0 ; 1]
Id_Puits_Climatique	Identifiant du puits climatique associé	-	Int	[1, +∞]
Is_Rafnoc	Présence d'une fonction de rafraîchissement nocturne associée au bouche-conduit	-	Int	[0, 1]
Param_Rafnoc_MiSaison	Paramètres pour le Rafraîchissement nocturne en mi-saison*	-	Double	[[-∞, +∞]]
Param_Rafnoc_Ete	Paramètres pour le Rafraîchissement nocturne en été*	-	Double	[[-∞, +∞]]
Pvent_rafnoc_rep	Puissance électrique du ventilateur de reprise en rafraîchissement nocturne	W	Double	[0 ; +∞]
Pvent_rafnoc_souf	Puissance électrique du ventilateur de soufflage en rafraîchissement nocturne	W	Double	[0 ; +∞]

(*) : Les paramètres du rafraîchissement nocturne sont (par ordre de saisie) :

- *Hleg_deb_tafnoc* : Heure légale de début de période de rafraîchissement nocturne.
- *Hleg_fin_tafnoc* : Heure légale de fin de période de rafraîchissement nocturne.
- *Theta_sd_dec_rafnoc* : Température intérieure de déclenchement du Rafnoc. en inoccupation nocturne associée au bouche-conduit
- *Theta_sd_arr_rafnoc* : Température intérieure d'arrêt du Rafnoc. en inoccupation nocturne associée au bouche-conduit.

- *Delta_Theta_lim_rafnoc* : Ecart limite entre les températures mesurées par les sondes intérieures et extérieures pour autoriser le rafraîchissement nocturne
- *Theta_ext_rafnoc* : Température extérieure limite pour autoriser le rafraîchissement

En cas d' un composant ventilation mécanique avec un calcul pré et post calcul hydrique, on donne à l' utilisateur la possibilité de faire uniquement le calcul de la partie post-calcul hydrique et de laisser le moteur faire le calcul de la partie pré-Hydrique dans le cas où ce dernier est strictement identique au calcul pour une ventilation double flux simple.

Pour cela, il faut que le jeu de donnée d' entrée hérite de l' interface « *IRT_Data_double_Flux* ». Cette interface est présente dans la librairie *RT21012.Extensions*.

5.6.4 L' INTERFACE DU CODE DE CALCUL

Le code de calcul d'une ventilation mécanique est défini par l'interface « IMoteur_Ventilation_Mecanique_Extension » et « IMoteur_Ventilation_Mecanique_Multi_Fonctions » si ce dernier est fait en deux parties.

	Non DAV	DAV
IMoteur_ventilation_Mecanique_Extension	X	X
IMoteur_Ventilation_Mecanique_Multi_Fonctions		X

La description des propriétés de l'interface « IMoteur_ventilation_Mecanique_Extension » est la suivante :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Ivent	Indicateur de ventilation de la zone (Occ / Inocc) du point de vu des scénarios	-	Int	[[0; +∞]]

Sorties				
Nom	Description	Unité	Type	Intervalle
Index	Identifiant unique d' un espace tampon	-	int	
Id_Et	Identifiant de l'espace tampon associé. Cette sortie n' est pas utilisée par le moteur	-	Int	[0; +∞]
H_vent	Déperditions par système de ventilation Bbio	-	double	[0; +∞]

T_Extrait	Température de l'air extrait de la ventilation mécanique	°C	Double	[-50; +∞]
Qm_Air_Extrait	Température de l'air extrait par la ventilation	Kg/s	Double	[0; +∞]
T_air_int_moy	Température intérieure barycentrique des groupes	°C	Double	[-50; +∞]
T_melange	Température après une boîte de mélange (si existante)	°C	Double	[-50; +∞]
Aut_Ch	Indicateur de saison de chauffage vue par distribution CTA	-	Int	[0; 1]
Aut_Fr	Indicateur de saison de refroidissement vue par distribution CTA	-	Int	[0; 1]
Q_Chaud	Besoins en Chaud d'une CTA	W	double	[0; +∞]
Q_Froid	Besoins en Froid d'une CTA	W	double	[0; +∞]
Q_Chaud_Antigel	Besoins en Chaud d'une CTA pour l'Antigel	W	double	[0; +∞]
Q_Chaud_Humidificateur	Besoins en Chaud d'une CTA pour l'humidificateur	W	double	[0; +∞]

L' interface contient aussi deux propriétés particulières. Ces dernières sont remplies par le moteur.

Nom	Description	Type
Ventil_Specifique_Soufflage_Collection	Collection des ventilations spécifiques connectées de reprise	List<IMoteur_Ventil_Specifique_Soufflage>
Ventil_Specifique_Reprise_Collection	Collection des ventilations spécifiques connectées de soufflage	List<IMoteur_Ventil_Specifique_Reprise>

Le développeur d' extension dynamique doit donc uniquement implémenter cette propriété de l' interface en créant un « getteur », et instancier la variable locale encapsulée dans le « getteur ». Le moteur se chargera ensuite de les remplir.

L' interface déclare également les méthodes « Init » et « Run » respectivement, appelées en début de calcul et à chaque pas de temps.

Pour les ventilations mécaniques sont le calcul se fait dans deux parties, l' utilisateur doit aussi implémenter l' interface « IMoteur_Ventilation_Mecanique_Multi_Fonctions ».

Cette interface contient en entrée la propriété :

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Ind_ByPass	Indicateur de l'activation ByPass par la partie I de la ventilation transmis à la partie II	-	Int	[[0; +∞]]

L' utilisateur doit aussi implémenter la méthode « *Calcul_Debits_PartieII* » où se trouve la partie II du calcul horaire.

Attention : Si le titre V concerne uniquement la partie II du calcul (présence de « *IRT_Data_Double_flux* » dans le data), seule la méthode « *Calcul_Debits_PartieII* » est implémentée. La méthode « *run* » doit avoir une implémentation vide.

5.7 LA GESTION ET REGULATION DU BALLON THERMOSTAT

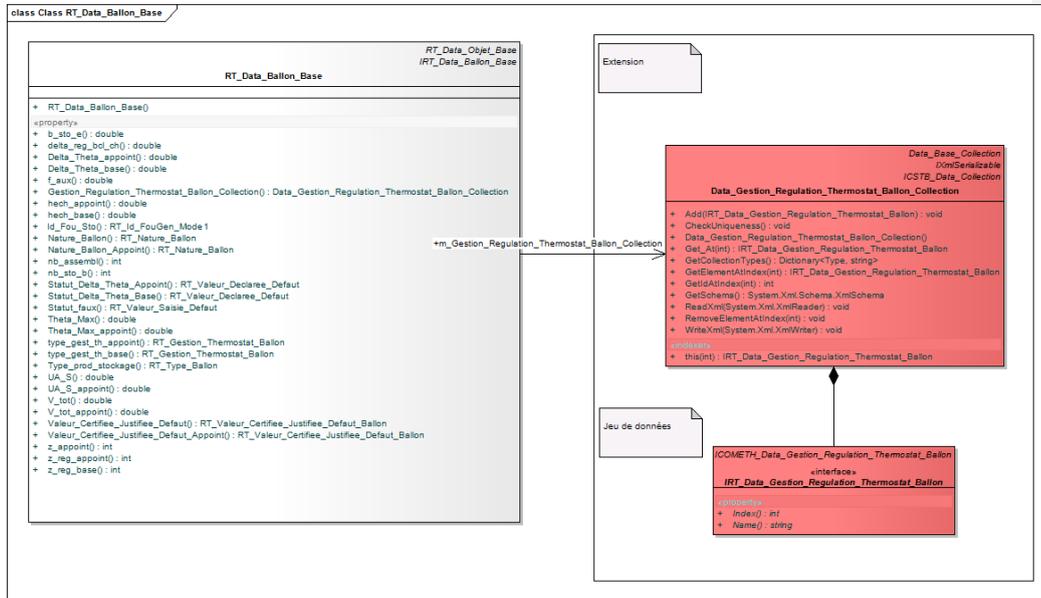
5.7.1 A PROPOS DE LA GESTION ET REGULATION DU BALLON THERMOSTAT

L' objet production stockage est utilisé pour la modélisation des ballons de stockage fournissant de l' eau chaude dédiée à l' ECS ou au chauffage. Un ou deux générateurs sont associés au ballon (le générateur de base et, le cas échéant, le générateur d' appoint).

Chacun des générateurs possède une sonde de température dans le ballon. Sa hauteur dans le ballon est à saisir par l' utilisateur. En fonction de cette température, et d' autres paramètres de régulation qui lui sont propres comme la température de consigne des ballons ou l' hysteresis de régulation, la gestion-régulation du ballon va transmettre au générateur l' information selon laquelle il doit fonctionner ou pas. Dans Th-BCE, tant que le ballon n' a pas atteint sa température de consigne, le générateur continuera à fonctionner.

La gestion régulation du thermostat du générateur de base est appelé, dans les calculs, avant la gestion-régulation du thermostat du générateur d' appoint.

5.7.2 IMPLEMENTATION DANS LE JEU DE DONNEES



5.7.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

Le jeu de paramètres d' entrée d' un composant production stockage est défini par l' interface : « **IData_Gestion_Regulation_Thermostat_Ballon_Extension** ».

Entrées du composant

Nom	Description	Unité	Type	Intervalle
Type_gestion_th_base	Type de gestion thermostat de base	-	RT_Gestion_Thermostat_Ballon_Extension	[[0; +2]]

5.7.4 L' INTERFACE DU CODE DE CALCUL

L'interface à implémenter pour la classe de calcul est la suivante : « **IMoteur_Gestion_Regulation_Thermostat_Ballon_Extension** ». Elle contient 2 fonctions à implémenter :

5.7.4.1 Calcul l'autorisation de fonctionnement

```
int Calcul_de_f_p();
```

5.7.4.2 Détermine si la zone recoit de la chaleur

```
bool Zone_Recoit_Chaleur(
```

```

int zone,
bool IsBase,
double theta_c,
double Rho_Cp_Eau,
double vp_tot,
int z_reg,
double delta_theta,
double[] theta_z_prec,
double[] theta_z_prec_h_2,
double[] v_z,
double[] p_e_z);

```

Description des paramètres de la fonction :

Nom	Description	Type
zone	Numéro de la zone	Int
IsBase	Si nous avons affaire à une base IsBase = True	Bool
theta_c	Température de consigne °C	Double
Rho_Cp_Eau	Masse volumique de l'eau liquide dans les conditions standards de pression et de température (P = 1atm, T=20°C) en (kg/m ³)	Double
vp_tot	Volume total puisé pendant le pas de temps	Double
z_reg	Numéro de la zone du ballon qui contient le système de régulation	Int
delta_theta	Hystérésis du thermostat (K)	Double
theta_z_prec	Température de la zone (z = m_z_a à 3) au pas de temps précédent (°C)	double[]
theta_z_prec_h_2	Température de la zone (z = m_z_a à 3) au pas de temps h - 2 (°C)	double[]
v_z	Volume de la zone z du ballon (z= m_z_a à 3)	double[]
p_e_z	Pertes thermiques de la zone z du ballon au pas de temps précédent	double[]

5.7.4.3 Paramètre particulier

L' interface contient aussi une propriété particulière. Cette dernière est remplie par le moteur.

Nom	Description	Type
Data	Donne accès à la variable Type_gestion_th_base	ICOMETH_Data_Gestion_Regulation_Thermostat_Ballon

Le développeur d' extension dynamique doit donc uniquement implémenter cette propriété de l' interface en créant un « getteur », et instancier la variable locale encapsulée dans le « getteur ». Le moteur se chargera ensuite de les remplir.

L' interface déclare également les méthodes « Init » et « Run » respectivement, appelées en début de calcul et à chaque pas de temps.

5.8 LES DISTRIBUTIONS RAFRAICHISSEMENT DIRECT

5.8.1 A PROPOS DES DISTRIBUTIONS RAFRAICHISSEMENT DIRECT

Une interface d'extension dynamique spécifique à l'ajout de systèmes de rafraichissement direct a été mise en place.

A la différence des systèmes de climatisation au sens de la méthode Th-BCE, les systèmes de rafraichissement peuvent contribuer partiellement et/ou ponctuellement au confort thermique des occupants, sans pour autant permettre le maintien avec suffisamment de garanties de la température de consigne souhaitée. De ce fait, ils ont la spécificité de devoir être pris en compte à la fois à calcul des consommations annuelles (Cep) et en calcul confort d'été (Tic ou mode Th-D). La prise en compte d'un système de rafraichissement se rapproche en cela de celle du puits climatique présent dans la méthode Th-BCE.

On vise l'ensemble des dispositifs de récupération directe d'une source froide par circulation d'eau ou air froids, incluant des échangeurs thermiques intermédiaires éventuels mais pas de groupe frigorifique.

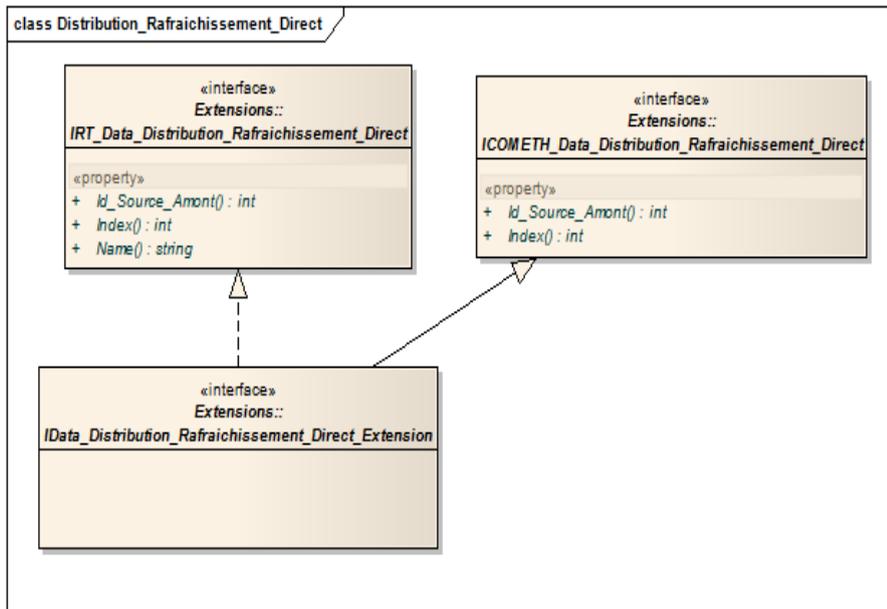
Un système de rafraichissement est généralement constitué d'un dispositif d'émission, d'un réseau de distribution (en général à deux niveaux, correspondant au groupe et à l'intergroupes), une composante échangeur et une composante « source » (assimilable à la notion de source amont).

C'est pourquoi le traitement du rafraichissement direct nécessite le recours à deux extensions dynamiques communiquant entre elles :

- une au niveau du groupe, **Emission_Rafraichissement_Direct**, décrite au chapitre 5.9, qui vise à modéliser les émetteurs de rafraichissement et calculer la puissance injectée dans le groupe. Un groupe peut intégrer au maximum une extension de type `Rafraichissement_Emission`.
- une au niveau génération, **Distribution_Rafraichissement_Direct**, décrite dans le présent chapitre, qui permet d'intégrer la régulation, la distribution entre groupes et les échangeurs de production d'eau froide. L'extension `Rafraichissement_Distribution` intègre une collection d'extensions `Rafraichissement_Emission` permettant ainsi de gérer le cas multi-groupes. Elle permet également la communication avec un composant « `Source_Amont` », afin de récupérer une température de source ainsi que des consommations de circulateurs amont.

5.8.2 IMPLEMENTATION DANS LE JEU DE PARAMETRES

Des distributions rafraichissement direct peuvent être rajoutées dans la collection `RT_Data_Distribution_Rafraichissement_Direct_Collection`.



5.8.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

Le jeu de paramètres d' entrée d' une distribution rafraichissement direct est défini par l' interface « `IData_Distribution_Rafraichissement_Direct_Extension` ».

La description des propriétés de l' interface « `IData_Distribution_Rafraichissement_Direct_Extension` » est la suivante :

Nom	Description	Unité	Type	Intervalle
Name	Nom de la source amont	-	String	-
Index	Identifiant unique	-	int	[0 ; +∞]
Id_Source_Amont	Identifiant de la source amont	-	int	

5.8.4 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d' une source amont est défini par l' interface « IMoteur_Distribution_Rafraichissement_Direct_Extension».

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Theta_Amont	Température de la source amont au pas de temps h-1	°C	double	
Waux_Am	Consommation électrique de source amont	Wh	double	
Emissions	Liste des emissions de rafraichissement direct associés		List< IMoteur_Emission_Rafraichissement_Direct>	
Sorties				
Nom	Description	Unité	Type	Intervalle
Phi_Rejet	Quantité d' énergie rejetée à la source amont au pas de temps h (positive ou négative)	W	double	
Pmax	Puissance nominal	W	double	
Taux_Charge	Taux de charge		double	

L' extension doit implémenter les fonctions de communication avec le moteur de calcul suivantes :

void **Run_OnNewHour()**

Cette fonction est appelée au début du pas de temps. Elle permet d' initialiser les variables internes et sorties, et d' intégrer des éléments de régulation avant d' appeler les fonctions comprises dans Emission_Rafraichissement_Direct.

Sorties à calculer : aucune

void **Run_Distribution()**

Son rôle est de calculer et communiquer une valeur d' énergie rejetée Φ_{Rejet} à la source amont. Φ_{rejet} a un impact sur la consommation d' auxiliaires de la source amont et sur la température amont du pas de temps suivant.

Sorties à calculer : Φ_{Rejet}

void **Run_Consommation()**

Cette fonction permet de finaliser les calculs de consommations de l' ensemble du système en rassemblant toutes les informations de l' ensemble des Emission_Rafraichissement_Direct. Elle doit inclure l' appel des fonctions Run_Consommation_Et_Pertes() de chaque Emission_Rafraichissement_Direct au travers d' une boucle sur la collection.

Elle a accès à la consommation d' auxiliaires de la source amont $W_{\text{aux_Am}}$ au pas de temps h.

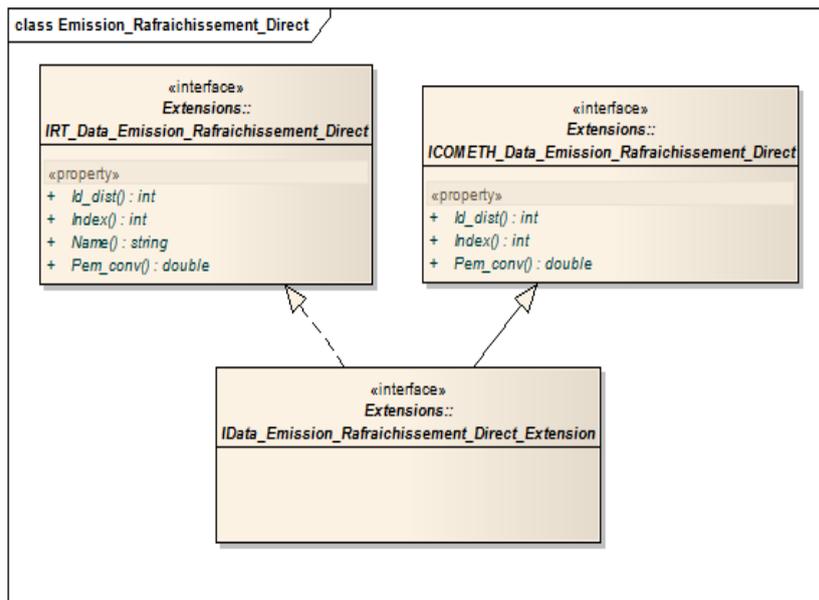
5.9 LES EMISSIONS RAFRAICHISSEMENT DIRECT

5.9.1 A PROPOS DES EMISSIONS RAFRAICHISSEMENT DIRECT

Voir §5.8.1.

5.9.2 IMPLEMENTATION DANS LE JEU DE PARAMETRES

Des émissions rafraichissement direct peuvent être rajoutées dans la collection `RT_Data_Emission_Rafraichissement_Direct_Collection`.



5.9.3 IMPLEMENTER L' INTERFACE DU JEU DE PARAMETRES D' ENTREE

Le jeu de paramètres d' entrée d' une emission rafraichissement direct est défini par l' interface « IData_Emission_Rafraichissement_Direct_Extension».

La description des propriétés de l' interface « IData_Emission_Rafraichissement_Direct_Extension» est la suivante :

Nom	Description	Unité	Type	Intervalle
Name	Nom de la source amont	-	String	-
Index	Identifiant unique	-	int	[0 ; +∞]
Pem_conv	Ratio d' énergie convective sur l' énergie totale transmise par l' émetteur de rafraichissement		double	
Id_dist	Index de la distribution rafraichissement direct	-	int	

5.9.4 IMPLEMENTER L' INTERFACE DU CODE DE CALCUL

Le code de calcul d'une source amont est défini par l'interface « IMoteur_Distribution_Rafraichissement_Direct_Extension ».

Entrées du composant				
Nom	Description	Unité	Type	Intervalle
Theta_sd	Température de sonde intérieure au sens du §10.1 de Th-BCE est passée en donnée entrée.	°C	double	
Theta_Cons_Fr	Température de consigne de refroidissement du groupe	°C	double	
IRelance	Indicateur de relance au temps h		int	
Aut_ch_eff	Autorisation de chauffage effective au pas de temps h	-	Int	
Theta_i_moy	Température d'air intérieur moyenne sans prise en compte des puissances émises sur le pas de temps h	°C	double	
Theta_rm_moy	Température radiante moyenne sans prise en compte des puissances émises sur le pas de temps h	°C	double	
W_int_fin	Humidité spécifique intérieure du groupe au pas de temps h-1	Kg/kg s	double	
Theta_ext	Température d'air extérieure au pas de temps h	°C	double	
W_ext	Humidité spécifique extérieur au pas de temps h	Kg/kg s	double	
Phi_crois	Puissance requise pour maintenir la température de consigne du rafraichissement direct au pas de temps h	W	double	
I_occ	Indicateur d'occupation au temps h		int	
Sorties				
Nom	Description	Unité	Type	Intervalle
Theta_Cons	Température de consigne du rafraichissement au pas de temps h	°C	double	
Is_On	Indicateur booléen d'activation du rafraichissement au pas de temps h	-	bool	

Phi_utile	Puissance de rafraichissement utile appliquée au groupe au pas de temps h	W	double
W_aux_dis	Consommation électrique du rafraichissement pour le poste « distribution »	Wh	double
W_aux_vent	Consommation électrique du rafraichissement pour le poste « ventilation »	Wh	double
Qth_recup_vc	Quantité d' énergie récupérable au pas de temps suivant (positif ou négatif) agglomérant l' ensemble des pertes	Wh	double

L' extension doit implémenter les fonctions de communication avec le moteur de calcul suivantes :

bool Est_Active()

Fonction appelée à chaque pas de temps qui permet d' intégrer la gestion/régulation du rafraichissement (saisonnière, horaire, en fonction des conditions extérieures...). Elle doit renvoyer une la valeur du booléen Is_On. Retourne : Is_On

void Calcul_Theta_Cons()

Fonction appelée à chaque pas de temps qui doit définir une valeur de la température de consigne pour le rafraichissement au pas de temps h Theta_Cons. Theta_Cons est bien une variable dynamique, car elle intègre l' éventuelle gestion en fonction de l' occupation ou des conditions de chaque pas de temps.

Sortie à calculer : Theta_Cons

void Run_Emission()

Fonction appelée à chaque pas de temps, qui calcule la puissance utile en fonction de la puissance requise pour atteindre la température de consigne propre au rafraichissement direct. Elle peut également intégrer des calculs relatifs à la consommations du système (ventilateurs ou circulateurs locaux). Pour calculer la puissance utile, il est nécessaire de disposer des températures de référence du local Theta_i_moy (air) et Theta_rm_moy (rayonnante moyenne) et de la puissance requise Phi_crois.

Sortie à calculer : Phi_utile

5.10 DONNEES PARTAGEES : DONNEES CLIMATIQUES

Nom	Description	Unité	Type	Intervalle
Mode	Mode de calcul en cours : Th_C = 1 ; Th_EB = 2 ; Th_EC = 3; Th_ERef = 4	Th_B = 0 ; -	int	[0 ; 4]
Vent	Vitesse du vent corrigée par le coefficient de rugosité des zones urbaines	m/s	double	[0 ; +∞]
Theta_ext	Température extérieure du site	°C	double	[-∞; +∞]
Wesite	Humidité extérieure d'air sec type	[kg/kg]	double	[0 ; 1]
Temp_Ciel	Température du ciel	°C	double	[-∞; +∞]
Teau	Température deau froide alimentant les systèmes de production d'ecs	°C	double	[-∞; +∞]
Hs	Gamma : Hauteur angulaire du soleil au-dessus de l'horizon	°	double	[-90 ; 90]
As	Azimuth du soleil (angle du soleil par rapport au sud - Positif vers l'Ouest - Négatif vers l'Est)	°	double	[-180 ;180]
Temp_Base	Température de base	°C	double	[-∞; +∞]
Rho_ext	Masse volumique de l' air extérieur	kg/m3	double	[0 ; +∞]

5.11 DONNEES PARTAGEES : DONNEES TEMPORELLES

Nom	Description	Unité	Type	Intervalle
T_Simul	Numéro du pas de temps en cours lors de la simulation	h	int	[0 ; +∞]
Duree_Simul	Durée totale de la simulation	h	Int	[0 ; +∞]
EHLO	Ecart heure locale et heure solaire	h	int	[1;2]
Htsmd	Heure de début du pas de temps solaire moyen	h	int	[0 ;8760[
Htsmf	Heure de fin du pas de temps solaire moyen	h	int]0 ;8760]
IHJ	Heure du Jour en temps solaire moyen	h	int	[1 ; 24]
IHS	Heure de la semaine temps solaire moyen	-	int	[1 ; 168]
IMOIS	Numéro du mois	-	RT_Mois	[1,12]
ISEM	Numéro de la semaine	h	int	[1 ; 53]
Jour_annee	Numéro du jour de l'année	-	int	[1; 365]
Jour_Mois	Jour du mois	-	int	[1; 31]
Jsem	Jour de la semaine	-	RT_Jour	[1 ; 7]
Heure_Legal	Heure Légale : Heure du jour + écart solaire	h	Int	[0 ; 24[
Heure_Locale_Semaine	Heure de la semaine avec correction solaire	h	int	[1 ;168]
Periode_Journee	Période de la journée (Matinée = 0, journée = 1, soirée = 2, nuit = 3)	h	RT_Periode_Journee	[0 ; 3]

5.12 METHODES MUTUALISEES ACCESSIBLES AU DEVELOPPEMENT D' UNE EXTENSION DYNAMIQUE

Des méthodes développées dans le cadre de la méthode Th-BCE sont mises à disposition du développeur de l' extension dynamique.

Toutes ces méthodes sont statiques sont disponibles dans différents noms d' espace.

5.12.1 CLASSE RT2012.EXTENSIONS.OUTILS_PREPRO_GENERATEUR

Elle contient les méthodes permettant de remplir les matrices de performances et de puissance absorbée des générateurs thermodynamiques avec les valeurs par défaut et l' ordre de remplissage spécifié par la méthode Th-BCE (sections 10.21 et 10.22).

Voir la documentation WEB sur <http://ese2.cstb.fr/Documentation/RT2012> pour les détails.

6. RECOMMANDATIONS DIVERSES:

6.1 LISTE DES BIBLIOTHEQUES (DLL) A REFERENCER :

Dans le cadre d' un développement d' un titre V, vous ne pouvez référencer que les bibliothèques suivantes :

- RT2012.Extensions.dll
- CSTB_Data_Lib.dll

 Le titre V ne doit pas référencer directement le moteur de calcul RT2012.dll, ceci créera une dépendance forte à ce dernier, et tout changement dans le moteur pourra impacter votre titre V et provoquer un dysfonctionnement de ce dernier.

6.2 SAISIE DES DONNEES SOUS FORME DE TABLEAUX (MATRICE) :

Pour des raisons de simplification de l' interface, le cœur de calcul RT2012 utilise les tableaux en entrée sous le format CSV, et se charge de les convertir avant l' utilisation; et vice versa pour l' écriture.

Exemple :

La matrice des performances est saisie sous la forme d' une chaîne de caractères (CSV), puis transformée en tableau au sein de l' objet **Data**.

Les méthodes de conversion seront mises à disposition dans la DLL **RT2012.Extensions**; en attendant ci-joint le code source de ces dernières.

```
[XmlIgnore]
[CSTB_Data_Property(Title = "Matrice des performances (COP ou EER)", Value = "0", Min = "0")]
[CSTB_Data_Matrix(IsColored = true)]
public double[,] Performance
{
    get{return M_Generateur.Performance; }
    set{M_Generateur.Performance = value; }
}

[XmlElement("Performance"), Browsable(false)]
[EditorBrowsable(EditorBrowsableState.Never)]
public string Performance_DTO
{
    get{return this.ConvertDouble2DArrayToString(Performance); }
    set{Performance = this.ConvertStringToDouble2DArray(value); }
}
/// <summary>
/// Permet de convertir une chaine de caractères en matrice de nombres flottants
/// </summary>
/// <param name="anArray">La chaîne de caractères</param>
/// <returns>Une matrice de double</returns>
protected double[,] ConvertStringToDouble2DArray(string anArray)
{
    string[] rows = anArray.Split(';');
    List<string[]> columns = new List<string[]>();

    foreach (string row in rows)
        columns.Add(row.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries));
    // Obtient la plus grande dimension des colonnes dans le cas de colonnes de tailles différentes
    int columnsDimension = (from column in columns
        select column.Count()).Max();
    double[,] result = new double[rows.Count(), columnsDimension];
    int rowIndex = 0;
    foreach (string[] column in columns)
    {
        int columnIndex = 0;
        foreach (string value in column)
        {
            result[rowIndex, columnIndex] = double.Parse(value.Trim().Replace(",", "."),
                System.Globalization.CultureInfo.InvariantCulture);
            columnIndex++;
        }
        rowIndex++;
    }
    return result;
}
/// <summary>
/// Permet de convertir une chaine de caractères en liste de nombres flottants
/// </summary>
/// <param name="anArray">La chaîne de caractères</param>
/// <returns>Une liste de double</returns>
protected double[] ConvertStringToDoubleArray(string anArray)
{
    string[] literalNumbers = anArray.Split(new char [] { ' ' },
        StringSplitOptions.RemoveEmptyEntries);
    double[] result = new double[literalNumbers.Length];
    int index = 0;
    foreach (string value in literalNumbers)
    {
        result[index] = double.Parse(value.Trim().Replace(",", "."),
            System.Globalization.CultureInfo.InvariantCulture);
        index++;
    }
    return result;
}
}
```

6.2 JEU DE PARAMETRES D' ENTREE DU SYSTEME TITRE V:

La modélisation d' un jeu de paramètres d' une extension dynamique ne doit pas contenir des types complexes, toutes les entrées doivent être explicitées dans le composant Data de l' extension dynamique.

Tous les paramètres du jeu de données d' entrée de l' extension dynamique doivent être déclarés comme des propriétés de l' objet Data.

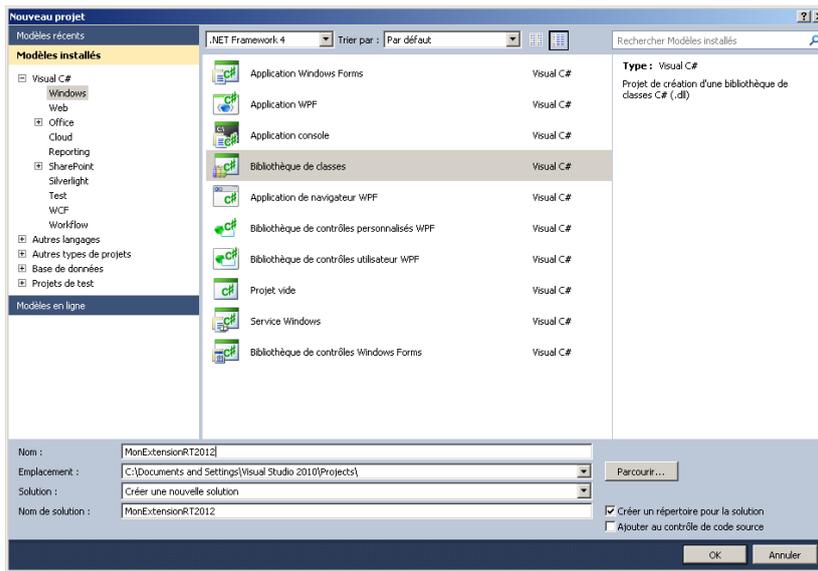
Les objets de types complexes ne sont pas autorisés en entrée de l' extension dynamique. En cas de modélisation avec un objet complexe, le jeu de paramètres d' entrée de ce dernier doit être explicité dans le jeu d' entrée principal de l' extension dynamique.

Les collections ne sont pas autorisées dans le jeu de paramètres d' entrée de l' extension dynamique. En cas d' utilisation de paramètres existants dans le jeu de donnée d' entrée principal du moteur, des paramètres synthétiques doivent être calculés, avant de les intégrer au jeu de donnée de l' extension dynamique.

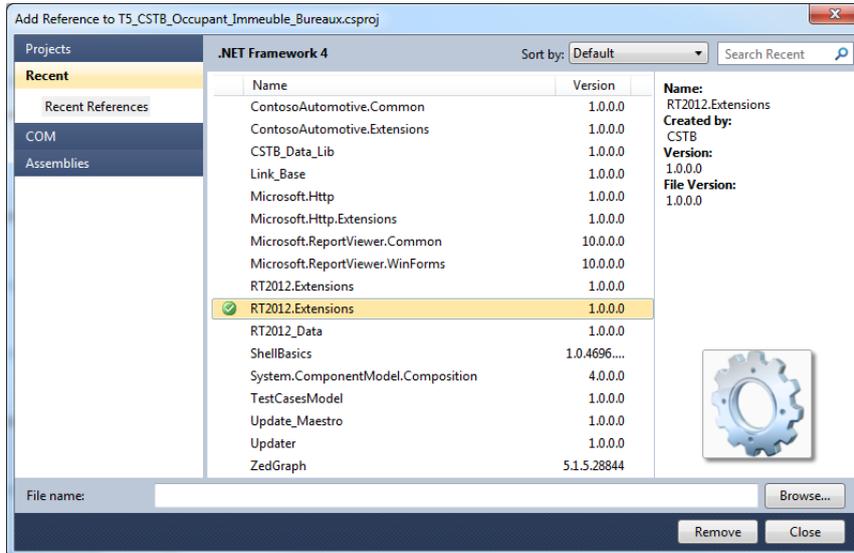
7. ANNEXES

7.1 METTRE EN PLACE UN PROJET D' EXTENSION DANS VISUAL STUDIO 2010

Pour concevoir une extension, un nouveau projet de type « Bibliothèque de classes » doit être créé, dans Visual Studio 2010.



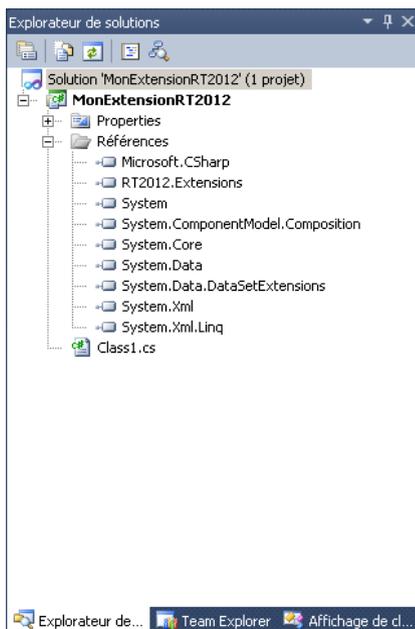
Ensuite, plusieurs références doivent être ajoutées au projet. Pour cela, cliquer droit sur le projet ou le répertoire « Références » de l'explorateur de solution et choisir « Ajouter une référence... ». Dans la fenêtre qui s'est ouverte, dans l'onglet .NET, choisir la librairie suivante « System.ComponentModel.Composition ».



Remarque : pour pouvoir utiliser cette librairie, le framework .Net 4.0 est nécessaire.

Ensuite, de la même façon, ajouter la librairie « RT2012.Extensions.dll », à partir de l'onglet « Parcourir » de la fenêtre « Ajouter une référence ».

L'explorateur de solutions doit ressembler à ceci maintenant :



A partir de ce point, l'extension peut être développée. Se référer à la section 2 pour les étapes détaillées, indépendamment de l'outil de développement utilisé.

7.2 MODELE DE FICHE ALGORITHME

Voir document joint, ModeleFicheAlgorithme.pdf.

8. MODIFICATIONS

Quoi	Quand	Version
<ul style="list-style-type: none"> Rajout du tableau de check list des actions à effectuer (section 2.5) Ajout des décorateurs du jeu de données Rajout d' un squelette de fiche algorithme Réorganisation complète de la section sur les interfaces Relecture, corrections diverses. 	17/05/2013	DEE/DIMN – 13.009RE
<ul style="list-style-type: none"> Ajout de la section espace tampon 	05/06/2013	DEE/DIMN – 13.009RE
<ul style="list-style-type: none"> Ajout de la section distribution intergroupe (section 5.4) Corrections des sorties calendrier Le paramètre Id_Gen de la distribution intergroupe chaud et froid est renommé en Id_Generation Le paramètre Type_Prim de la distribution intergroupe chaud est renommé en Type_Prim_Chaud Le paramètre type_reseau_intergroupe_ecs de la distribution intergroupe ECS est renommé en Type_Reseau_Intergroupe_ECS 	19/06/2013	DEE/DIMN – 13.009RE
<ul style="list-style-type: none"> Ajout de la section productions stockage (section 5.5) 	09/07/2013	DEE/DIMN – 13.009RE
Ajout de la section Recommandations de codage (Section 6)	08/08/2013	DEE/DIMN – 13.009RE
Modifications de l' ouverture sur production_stockage. Afin de laisser libre court à l' imagination des développeurs de titres V, l' ouverture est maintenant générique. Elle permet la modélisation de n' importe quel composant (échangeurs, stockages, régulation...) venant entre le niveau génération et les générateurs, qu' il soit mono ou double service chauffage/ECS. Il n' est plus nécessaire de rapprocher le titre V de l' une	28/11/2013	DEE/DIMN – 13.009RE

des catégories de production_stockage ECS (base seule, etc...) prévue par la méthode Th-BCE.

Correction de la sortie moteur mode de calcul	28/11/2013	DEE/DIMN – 13.009RE
---	------------	------------------------

Ajout de la section Source Ballon Boucle solaire ECS /Chauffage (Section 5.1)	28/11/2013	DEE/DIMN – 13.009RE
--	------------	------------------------

Correction de la prise en compte des consommations ECS pour les générateurs extensions.	30/01/2014	DEE/DIMN – 13.009RE
---	------------	------------------------

Gestion et régulation du ballon thermostat	30/03/2014	DEE/DIMN – 13.009RE
--	------------	------------------------

Pac Triple service	01/04/2014	DEE/DIMN – 13.009RE
--------------------	------------	------------------------

Les Distributions Rafrachissement Direct	09/11/2016	DEE/DIMN – 13.009RE
--	------------	------------------------

Les Emissions Rafrachissement Direct
