



HAL
open science

Interopérabilité basée sur les standards Modelica et composant logiciel pour la simulation énergétique des systèmes de bâtiment

Sana Gaaloul

► **To cite this version:**

Sana Gaaloul. Interopérabilité basée sur les standards Modelica et composant logiciel pour la simulation énergétique des systèmes de bâtiment. Energie électrique. Université de Grenoble, 2012. Français. NNT : . tel-00782540

HAL Id: tel-00782540

<https://theses.hal.science/tel-00782540>

Submitted on 30 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Génie électrique**

Arrêté ministériel : 7 août 2006

Présentée par

« **Sana GAALOUL CHOUIKH** »

Thèse dirigée par « **Frédéric WURTZ** » et
codirigée par « **Benoit DELINCHANT** »

préparée au sein du **Laboratoire de Génie Électrique de
Grenoble (G2ELAB)**
dans l'**École Doctorale d'Électronique, Électrotechnique,
Automatique et Traitement de Signal (EEATS)**

Interopérabilité basée sur les standards Modelica et composant logiciel pour la simulation énergétique des systèmes de bâtiment

Thèse soutenue publiquement le «**18/10/2012**»,
devant le jury composé de :

Mr Etienne WURTZ

Directeur de recherche au CEA/INES, Président

Mr Claude MARCHAND

Professeur à l'Université de Paris Sud, Rapporteur

Mr Xavier ROBOAM

Directeur de recherche à l'Université de Toulouse, Rapporteur

Mr Micheal WETTER

Chercheur au Laboratoire National de Lawrence Berkeley, Membre

Mr Patrick BEGUERY

Ingénieur R&D de Schneider-electric, Membre

Mr Frédéric WURTZ

Directeur de recherche à l'INP Grenoble, Directeur de thèse

Mr Benoit DELINCHANT

Maître de conférence à l'UJF, Co-directeur de thèse



Remerciements

Tout d'abord, je tiens à remercier mes encadrants qui m'ont offert l'opportunité de travailler sur un sujet aussi transversal que celui-ci. Cette thèse n'aurait pas pu avoir lieu sans un encadrement de très grande qualité que celui de M. Frédéric Wurtz et M. Benoit Delinchant. J'ai eu beaucoup de chance d'être encadrée par deux personnes aussi complémentaires.

- Frédéric Wurtz, mon directeur de thèse, pour sa passion à la recherche qu'il savait transmettre, son ouverture d'esprit, son esprit de synthèse admirable et surtout sa capacité exceptionnelle de motivation. Il a toujours su trouver les mots pour me remonter le moral même pendant les périodes les plus dures.

Fred, heureusement que tu n'avais pas noté tout ce que tu as dit «rappeler le jour de ma soutenance»☺.

- Benoit Delinchant, pour son sens critique, pour sa disponibilité et pour son aide précieuse dont j'ai bénéficié durant ces trois années de thèse. Grâce à son soutien et sa confiance, ma thèse s'est déroulée dans des bonnes conditions.

J'ai pris énormément de plaisir en travaillant avec toi sur le niveau professionnel ainsi que personnel.

J'adresse ensuite mes remerciements aux membres de mon jury:

- M. Etienne Wurtz : directeur de recherche au CEA/INES et président de l'association IBPSA-France pour m'avoir fait l'honneur de présider mon jury de thèse et pour son investissement dans la communauté de simulation du bâtiment essentiellement à travers l'organisation des conférences et des écoles thématiques.

- M. Xavier Roboam: directeur de recherche à l'Université de Toulouse et M. Claude Marchand: professeur à l'Université de Paris Sud, d'avoir accepté de rapporter ce travail ainsi que pour leurs critiques et remarques intéressantes et enrichissantes.

- M. Micheal Wetter: chercheur au Laboratoire National de Lawrence Berkeley, pour l'intérêt qu'il a porté à mon travail ouvrant des perspectives de collaboration.

- M. Patrick Beguery: ingénieur R&D de Schneider-electric, d'avoir apporté une vision industrielle à ce travail de recherche.

J'ai eu la chance de travailler au sein d'un grand laboratoire, le G2elab dont je tiens à remercier tous les permanents qui ont veillé à assurer des bonnes conditions de travail :

chercheurs (dont spécialement ceux de mon équipe MAGE : Gérard, Jean Louis, Gilles, Olivier...), ingénieurs, service informatique et personnels administratifs. Une mention spéciale à Laurent Gerbaud pour ses conseils précieux ainsi qu'à Franck Verdière pour son aide.

J'ai aussi côtoyé des amis formidables dont la liste est bien longue et je veux en citer que quelques uns et j'espère que les autres se reconnaîtront: Mathieu, Didier, Fanny, Ni, Christophe, Yasmine, Michael, Ando, Ali, Hajer, Manel, Adrien, Bill, Ayesha..., ainsi que mes voisins, habitants de la salle Predis, anciens et nouveaux : Hoang Anh, Ghaith, Jonathan, Shadi, Ardavan, Franky, Kaustav et Mathilde. Merci pour l'ambiance et la convivialité que vous avez créés dans cet espace de travail et d'échange.

Une dédicace spéciale à mes deux copines tunisiennes Rim et Abir, où la mafia tunisienne comme on préfère nous appeler ☺...Rim et sa petite famille, pour les bons moments passés ensemble et Abir pour tout l'échange et partage personnels et professionnels qu'on a eu durant ces trois ans.

A ma chère Sarra (une sœur au vrai sens du terme) de m'avoir supportée depuis plus de 6 ans. Merci d'être toujours là à mes côtés pendant les périodes difficiles, également pour tous les beaux souvenirs qui resteront à jamais gravés dans ma mémoire.

Les mots m'échappent pour remercier mes deux familles Gaaloul et Chouikh pour leurs soutiens inconditionnels. Merci à ceux qui ont fait le déplacement pour assister à la soutenance, votre présence m'a fait chaud au coeur.

Une pensée particulière à mes trois frères Sami, Hedi et Karim. Du tout profond de mon cœur, je vous souhaite plein, plein de bonheur.

Ces remerciements n'auront pas de valeur sans une très grande pensée à mes chers parents. Ce manuscrit n'est certes pas à la hauteur de ma reconnaissance, mais je tiens à vous le dédier. Merci pour votre confiance et soutien interminables sans lesquels je ne serais pas arrivée jusque là. Que dieu vous protège et que vous trouvez ici mon amour éternel.

Et pour finir en beauté, je tourne mes pensées vers mon cher époux Mahdi pour lui exprimer ma gratitude pour sa patience et ses encouragements, particulièrement dans les derniers mois de cette aventure. Merci d'être toujours à mes côtés pour m'épauler et me donner la force pour avancer et aller toujours plus loin.

Table de matière

INTRODUCTION GÉNÉRALE	1
CHAPITRE 1: CONTEXTE ÉNERGÉTIQUE, PROBLÉMATIQUES ET ÉTAT DE L'ART DES SOLUTIONS EXISTANTES D'INTEROPÉRABILITÉ DE LA SIMULATION DYNAMIQUE DU BÂTIMENT	5
I. Introduction	7
II. Contexte énergétique et environnemental.....	8
II.1. Le bâtiment : un enjeu énergétique majeur	8
II.1.1. Le bâtiment : un secteur polluant et consommateur d'énergie.....	8
II.1.1.1. Au niveau mondial	8
II.1.1.2. Au niveau de la France	9
II.1.2. Le bâtiment : une charge électrique importante	10
II.1.2.1. Le bâtiment est le premier secteur consommateur d'énergie électrique	10
II.1.2.2. L'électricité est le premier vecteur d'énergie consommé dans le secteur du bâtiment	11
II.1.2.3. Conséquences de cette charge importante.....	12
II.1.3. Le bâtiment : une variable d'ajustement du réseau électrique	13
II.1.3.1. Le bâtiment est un acteur du réseau électrique	13
II.1.3.2. Lisser les courbes de consommations et réduire les pics de consommations	13
II.1.3.3. Diminuer les « black-outs »	15
II.1.4. Conclusion.....	16
II.2. Des nouvelles actions et résolutions dans le secteur du bâtiment	17
II.2.1. Motivations des efforts réalisés dans le secteur du bâtiment	17
II.2.2. Les bâtiments basses consommations, voire à énergie positive.....	17
II.2.2.1. Amélioration des performances énergétiques	17
II.2.2.2. Conséquences de ces évolutions	19
II.2.3. Les bâtiments intelligents (BI):.....	20
II.2.4. Conclusion.....	22
II.3. Des nouveaux besoins en simulation du bâtiment	23
II.3.1. Les intérêts de la simulation du bâtiment.....	23
II.3.2. Les nouveaux besoins de simulation du bâtiment.....	24
III. Problématiques actuelles de la simulation dynamique du bâtiment:	26
III.1. Introduction	26
III.2. Hétérogénéité des modèles.....	27
III.2.1. Différentes natures temporelles.....	27
III.2.2. Différentes approches de modélisation	28
III.2.3. Différents niveaux de finesse	29
III.2.4. Différents dimensions spatiales.....	31
III.2.5. Les difficultés liées à cette hétérogénéité et les besoins d'interopérabilité ..	31
III.3. Problématiques liées aux simulateurs	33
III.3.1. La multiplicité des outils de simulations.....	33
III.3.2. La non adéquation d'outils de simulation à de nouveaux besoins	34
III.3.3. La spécialisation des outils de simulations	35

III.3.3.1. Limitations des outils existants	35
III.3.3.2. Illustrations au travers des outils étudiés.....	35
III.3.4. Besoins d'interopérabilité entre les outils de simulation du bâtiment	37
IV. Les solutions existantes d'interopérabilité.....	40
IV.1. Stratégies d'interopérabilité :	40
IV.1.1. Interopérabilité des données des modèles	40
IV.1.2. Interopérabilité des processus des modèles.....	40
IV.1.3. Coopération des données et des processus des modèles	41
IV.2. Capacités d'interopérabilité des outils utilisés	42
IV.3. Conclusion.....	42
V. Conclusion.....	44
CHAPITRE 2 : LA DIVERSITÉ DES MODÈLES RENCONTRÉS DANS LE BÂTIMENT ET LES DIFFICULTÉS DE MODÉLISATION : ILLUSTRATION SUR LE CAS DE LA PLATEFORME PREDIS MHI.....	45
I. Introduction	47
II. Description du bâtiment PREDIS MHI	48
II.1. Objectifs du bâtiment PREDIS MHI.....	48
II.1.1. Le centre PREDIS « Smart networks for energy »	48
II.1.2. La plateforme PREDIS MHI.....	48
II.2. Description architecturale	49
II.3. Equipements	50
II.3.1. Une VMC double flux.....	50
II.3.2. Système de supervision et instrumentations expérimentales	51
II.3.3. Autres équipements	53
III. Modélisation de l'enveloppe thermique	54
III.1. Le modèle de l'enveloppe dans COMFIE Pleiades	54
III.1.1. Description et hypothèses du modèle.....	54
III.1.2. Traitement du modèle par le cœur de calcul de COMFIE	55
III.2. Le modèle réduit de l'enveloppe.....	56
III.2.1. L'intérêt du modèle réduit.....	56
III.2.2. Description du modèle réduit.....	57
III.2.3. Procédure d'identification des paramètres du modèle	58
III.3. Conclusion.....	60
IV. Modélisation des équipements : VMC double flux	61
IV.1. La partie électrique : Moteur et variateur de vitesse	61
IV.1.1. La machine asynchrone.....	61
IV.1.2. Le variateur de vitesse.....	62
IV.1.2.1. Les pertes du circuit d'électronique de puissance	62
IV.1.2.2. La loi de commande	63
IV.2. Les échangeurs de chaleur.....	64
IV.2.1. Le principe de modélisation	64
IV.2.2. La procédure de validation des modèles.	66
IV.2.3. L'échangeur rotatif.....	68
IV.2.3.1. Caractéristiques du modèle	68

IV.2.3.2. Implémentation et validation du modèle.....	70
IV.2.4. La batterie d'eau chaude (BEC).....	71
IV.2.4.1. Caractéristiques du modèle.....	71
IV.2.4.2. Implémentation et validation du modèle.....	72
IV.3. Le circuit aéraulique.....	73
V. Modélisation des occupants.....	76
V.1. Les approches de modélisation du comportement des occupants.....	76
V.1.1. Les intérêts de la modélisation multi-agent des occupants.....	77
V.1.2. Le modèle de l'occupant dans Brahms.....	77
VI. Les difficultés rencontrées lors de la modélisation d'un système bâtiment.....	80
VII. Conclusion.....	81
CHAPITRE 3 : LES APPROCHES DE MODÉLISATION SYSTÈME: APPLICATION AUX SYSTÈMES DE PREDIS.....	83
I. Introduction.....	85
II. L'approche « boîte blanche » de modélisation système se basant sur le langage Modelica.....	87
II.1. L'approche « boîte blanche » de modélisation système.....	87
II.2. La modélisation dans Modelica.....	88
II.2.1. Les principales caractéristiques du langage.....	88
II.2.1.1. Un langage multi-physique.....	88
II.2.1.2. Un langage basé sur des équations et orienté objet.....	88
II.2.1.3. Une représentation causale et/ou acausale.....	89
II.2.2. La structure des modèles en Modelica.....	90
II.2.2.1. La structuration d'un modèle.....	90
II.2.2.2. Les connexions entre les modèles.....	91
II.3. L'approche d'interopérabilité se basant sur le langage Modelica en application à la modélisation du système de bâtiment PREDIS.....	92
II.3.1. Description des modèles de PREDIS dans Modelica.....	92
II.3.1.1. Modélisation de la VMC dans Modelica.....	92
II.3.1.2. Modélisation de l'enveloppe et des usages dans Modelica.....	95
II.4. Les limites de cette approche d'interopérabilité.....	98
III. L'approche boîte noire se basant sur le standard ICAr de composants logiciels ..	100
III.1. Le paradigme de composants logiciels.....	100
III.1.1. Le composant logiciel.....	100
III.1.2. Le standard de composants logiciels : Caractéristiques et concepts dérivants	101
III.1.2.1. Caractéristiques.....	101
III.1.2.2. Les types de composants logiciels.....	102
III.1.2.3. Un plug-out de composant logiciel.....	103
III.1.2.4. Un plug-in de composant logiciel.....	104
III.1.2.5. Un bus à composants logiciels.....	104
III.2. Le standard « ICAr » de composants logiciels.....	105
III.2.1. Spécificités de la norme ICAr.....	105
III.2.2. Les facettes de l'ICAr.....	107

III.3.	Les plug-ins et les plug-outs autour du standard ICAr.....	109
III.3.1.	Les plug-ins du standard ICAr.....	109
III.3.1.1.	Généralités sur la mise en œuvre.....	109
III.3.1.2.	Le plug-in Matlab / Simulink.....	111
III.3.1.3.	Le plug-in de l'ICAr vers Dymola.....	112
III.3.2.	Les plug-outs vers le standard ICAr.....	113
III.3.2.1.	Généralités.....	113
III.3.2.2.	Le plug-out Modelica.....	114
III.3.2.3.	Le plug-out de Brahms.....	114
III.3.2.4.	Le plug-out depuis COMFIE.....	115
III.3.3.	Limites et perspectives d'améliorations des plug-ins et des plug-outs.....	115
III.4.	Synthèse sur la complémentarité des deux approches d'interopérabilité.....	118
IV.	Conclusion.....	119
CHAPITRE 4 : LA SIMULATION DYNAMIQUE DES SYSTÈMES HÉTÉROGÈNES BASÉE SUR L'APPROCHE À COMPOSANTS LOGICIELS.....		121
I.	Introduction.....	123
II.	La simulation à pas variable des systèmes hybrides.....	124
II.1.	Introduction.....	124
II.2.	Architecture d'un cas d'étude : couplage entre un modèle d'enveloppe thermique (issu de COMFIE) et un modèle de chauffage (Matlab).....	125
II.3.	L'orchestration des modèles.....	127
II.3.1.	La simulation à pas fixe.....	127
II.3.1.1.	La simulation à pas fixe d'une heure.....	127
II.3.1.2.	La simulation à pas fixe de 6 minutes.....	127
II.3.2.	La simulation à pas variable.....	128
II.3.2.1.	Le principe de la simulation à pas variable.....	128
II.3.2.2.	Les adaptations des natures des modèles nécessaires pour une simulation à pas variable.....	129
II.3.2.3.	Résultats de la simulation à pas variable.....	131
II.4.	Exploitation de l'approche composant logiciel pour la simulation dynamique hybride.....	133
II.4.1.	La facette de simulation dynamique des modèles ODE de l'ICAr.....	133
II.4.1.1.	Les ports de la facette :.....	133
II.4.1.2.	Les interfaces de la facette.....	134
II.4.2.	Application de l'approche composant logiciel au cas d'étude.....	135
II.4.2.1.	La démarche de mise en place.....	135
II.4.2.2.	Les résultats de la simulation.....	136
II.4.3.	Intérêt de l'application de cette approche composant.....	137
II.5.	Conclusion.....	139
III.	La co-simulation des systèmes hétérogènes.....	140
III.1.	Introduction.....	140
III.1.1.	Les intérêts de la co-simulation.....	140
III.1.2.	Typologie des formes de co-simulation.....	140
III.1.3.	Intérêts de la co-simulation à base de composants logiciels.....	141
III.2.	Architecture du cas d'étude: Couplage entre COMFIE et Brahms dans Simulink.....	143

III.2.1.	Description du cas d'étude	143
III.2.2.	les intérêts de ce couplage	143
III.3.	Orchestration des acteurs de la co-simulation : exemples de stratégies d'orchestration.....	146
III.3.1.	La synchronisation à pas fixe	146
III.3.2.	La synchronisation à pas variable	146
III.3.2.1.	La synchronisation avec un retour en arrière	147
III.4.	Exploitation de l'approche composant logiciel pour la co-simulation.....	148
III.4.1.	La facette de co-simulation de l'ICAr.....	148
III.4.2.	Application de l'approche composant logiciel au cas d'étude.....	149
III.4.2.1.	La démarche de mise en place.....	149
III.4.2.2.	Résultats de la co-simulation.....	151
III.4.3.	Les intérêts de la co-simulation par l'approche composant	152
III.5.	Conclusion.....	154
IV.	Conclusion.....	155
	CONCLUSIONS ET PERSPECTIVES	157
	Conclusions	159
	Perspectives autour de l'approche composant logiciel	161
	BIBLIOGRAPHIE	163
	ANNEXES	181

Table des figures

Figure 1. Evolution du taux de concentration de différents gaz à effet de serre.....	8
Figure 2. Evolution de la consommation d'énergie primaire mondiale (en Mtep) selon la source [IEA 2011].	8
Figure 3. Répartition de la consommation d'énergie par secteur en France [SOES 2010].....	9
Figure 4. Les efforts réalisés par la France depuis 1997 pour améliorer l'efficacité énergétique des bâtiments (directives DPE) [ADEME 2010]	10
Figure 5. Répartition de la consommation électrique en France par secteur en 2010 [CGDD 2010].....	10
Figure 6. L'évolution de la consommation électrique en France corrigée des variations climatiques par secteur [CGDD 2010]	11
Figure 7. Répartition des sources d'énergie consommée dans le secteur résidentiel- tertiaire français en 2010 [ADEME 2011].....	11
Figure 8. Evolution des consommations unitaires des résidences principales par usage [ADEME 2010].....	12
Figure 9. Les acteurs du réseau électrique Français.....	13
Figure 10. L'évolution des puissances électriques maximales consommées depuis 2001 en fonction des températures [RTE 2011]	14
Figure 11. Consommations typiques journalières en été, en demi-saison (DS) et en hiver [RTE 2011].....	14
Figure 12. Trajectoire probable des risques de défaillance [RTE 2011].....	16
Figure 13. Les motivations pour l'amélioration de l'habitat [ADEME 2010].....	17
Figure 14. L'évolution des exigences réglementaires de la consommation énergétique des bâtiments neufs : une rupture opérée par le Grenelle Environnement [RT 2012]	18
Figure 15. Illustration des apports internes	19
Figure 16. Quantification de la part des apports internes dans la contribution au chauffage d'une salle du bâtiment PREDIS [CHEN 2010]	20
Figure 17. Les domaines interagissant dans le système bâtiment [SIMINTECH 2009].....	25
Figure 18. Représentation temporelle continue (gauche) et discrète (droite)	27
Figure 19. Approche analytique / approche empirique	28
Figure 20. Approche boîte blanche / boîte noire / boîte grise	28
Figure 21. Approche orientée (gauche) / approche non orientée (droite)	29
Figure 22 L'erreur dans la performance de prédiction vis-à-vis de la complexité du modèle et la notion de compromis [HEN 2012] [TRC 2010].....	30
Figure 23 Les dimensions spatio-temporelles.....	31
Figure 24. Les outils de BPS développés entre 1997 et 2010 [ATT 2011].....	33
Figure 25. Un aperçu du bâtiment PREDIS	49
Figure 26. L'espace bureau et la salle informatique de la plateforme PREDIS MHI.....	50
Figure 27. Le principe de la ventilation simple flux (gauche) double flux (droite) [LID 1966]	51
Figure 28. Composition de la VMC DF de PREDIS	51
Figure 29. L'interface de la GTC de supervision du premier étage.....	52
Figure 30. Les capteurs utilisés pour la modélisation de la VMC	53
Figure 31 Vue de dessus des zones modélisées dans Alcyone (module de COMFIE Pleaides)	54
Figure 32 L'analogie thermique électrique	57
Figure 33 Le schéma électrique équivalent de l'enveloppe thermique	57
Figure 34 L'algorithme d'identification.....	58
Figure 35 Comparaison entre le modèle avec les paramètres initiaux et la simulation	59

Figure 36 Comparaison entre le modèle avec les paramètres optimisés et la simulation	59
Figure 37 Modèle général des éléments de la partie électrique	61
Figure 38. Schéma équivalent du moteur asynchrone.....	62
Figure 39. Composition du circuit d'électronique de puissance du variateur de vitesse	63
Figure 40. Fonctionnement en V/f couple variable.....	64
Figure 41. Les régimes de circulation de l'air dans les échangeurs de chaleur	65
Figure 42. Les capteurs utilisés pour la modélisation de la VMC	67
Figure 43. Différence de mesure des températures entre l'entrée (capteur Tane) et la sortie de la VMC (mesurées au niveau du soufflage dans la salle informatique: capteur T-soufflage_Info).....	68
Figure 44. Les mesures de températures corrigées	68
Figure 45. L'échangeur rotatif (image réelle : gauche et schéma : droite)	69
Figure 46. La validation du modèle de l'échangeur rotatif	71
Figure 47. Le modèle de la batterie d'eau chaude.....	71
Figure 48. La validation du modèle de la batterie d'eau chaude.....	72
Figure 49. Des composants du circuit aéraulique de PREDIS	73
Figure 50. Les différentes pertes du ventilateur	74
Figure 51. Couple de résistance en fonction de la vitesse de rotation	75
Figure 52. La diversité du profil horaire pour une charge d'occupation typique d'un environnement de type bureau [WINK 1993].....	76
Figure 53. Le concept des agents dans la modélisation multi-agentw	77
Figure 54. Les éléments de modélisation de l'occupant dans Brahms [KAS 2011].....	78
Figure 55. L'évolution du comportement d'un occupant dans Brahms.	79
Figure 56. Exemple d'héritage dans Modelica.....	89
Figure 57. Etapes de compilation d'un code Modelica.....	90
Figure 58. La structure d'un modèle de capacité thermique dans Modelica.....	91
Figure 59. Les connexions dans Modelica	91
Figure 60. Modélisation causale de la partie électrique de la VMC dans Modelica.....	92
Figure 61. Résultats de simulation du modèle causal de la MAS de la VMC dans Modelica.	93
Figure 62. Modélisation acausale de la partie électrique de la VMC dans Modelica	93
Figure 63. L'adaptation du modèle de l'échangeur en Modelica pour supporter la rotation... ..	94
Figure 64. Projection du modèle de l'échangeur rotatif dans Modelica avec mise en oeuvre dans Dymola.....	94
Figure 65. Le modèle de la VMC de PREDIS dans Modelica.....	95
Figure 66. Résultats de la projection des modèles de la VMC dans Modelica	95
Figure 67. Projection du modèle de l'enveloppe et des usages dans Modelica	96
Figure 68. Le couplage entre la VMC et l'enveloppe du bâtiment PREDIS dans Modelica....	97
Figure 69. La complémentarité boîte noire/boîte blanche.....	99
Figure 70. La souplesse des composants logiciels [DEL 2011].....	100
Figure 71. Vue externe d'un composant logiciel	100
Figure 72. Les dimensions de la normalisation du composant logiciel	102
Figure 73. Le composant-modèle.....	103
Figure 74. Le composant-simulateur.....	103
Figure 75. Le concept de plug-out	104
Figure 76. Le concept de plug-in	104
Figure 77. Le concept de bus à composants logiciels	105
Figure 78. L'interface principale « component » de l'ICAr	107
Figure 79. Le principe « multi-facettes » de l'ICAr.....	107
Figure 80. Les facettes dynamiques pouvant être contenues dans un ICAR et leur hiérarchie (les fonctionnalités en rouge ne sont pas encore intégrées)	108

Figure 81. Exemple d'utilisation des plug-ins et des plug-outs de l'ICAr : réalisation d'un processus de simulation complet appliqué à la plate-forme PREDIS	109
Figure 82. Illustration des principales fonctionnalités d'un plug-in de la facette ODE de l'ICAr	110
Figure 83. Le plug-in Matlab/Simulink.....	111
Figure 84. Le plug-in Dymola.....	112
Figure 85. Les étapes du plug-in Dymola	112
Figure 86. Exemple du code Modelica automatiquement généré (ICAr ODE de COMFIE)	113
Figure 87. Le principe du plug-out d'un modèle Modelica (via JModelica)	114
Figure 88. Le principe du plug-out de Brahms	115
Figure 89. Le bus à composant logiciel développé autour de l'ICAr [DEL 2012C]	116
Figure 90. Le principe de composant composite.....	118
Figure 91. Schéma global de la simulation dans Simulnk	126
Figure 92. Résultats de la simulation, à pas fixe d'une heure, avec dépassement de seuil géré	127
Figure 93. Résultats de la simulation à pas fixe de 6 minutes, permettant une gestion fine de la régulation, mais qui requière un temps de simulation long.....	128
Figure 94. Le principe de la simulation à pas variable dans SIMULINK.....	129
Figure 95. Les ports et les méthodes typiques d'un modèle ODE hybride.....	129
Figure 96. Les oscillations de la puissance et la température dues à une variation brutales des entrées.....	130
Figure 97. Le modèle de chauffage du deuxième ordre.....	131
Figure 98. Les résultats de la simulation à pas variable comparées à ceux à pas fixe de 6 minutes	131
Figure 99. Les ports des entrées et des sorties du composant ICAr pour la simulation dynamique de type ODE	133
Figure 100. Le diagramme UML de la facette de simulation dynamique ODE de l'ICAr	134
Figure 101. La démarche de réalisation de la simulation du cas test en passant par l'approche ICAr.....	135
Figure 102. Implémentation de l'interface ODE de l'ICAr COMFIE	136
Figure 103. L'interaction entre l'ICAr encapsulé dans une S-fonction et le solveur de Matlab lors d'une simulation ODE continue.....	136
Figure 104. Les résultats de simulation de deux systèmes avec et sans passage par l'approche composants logiciels	137
Figure 105. Comparaison des résultats de simulation avec importation de l'ICAr-COMFIE dans Matlab/Simulink et Dymola.....	138
Figure 106. La co-simulation directe (gauche) et la co-simulation indirecte (droite).....	140
Figure 107. La co-simulation sans et avec l'utilisation d'un standard de composant logiciel.	142
Figure 108. Le schéma de la co-simulation entre COMFIE et BRAHMS orchestrée dans SIMULINK.	143
Figure 109. La synchronisation à pas fixe entre COMFIE et BRAHMS.....	146
Figure 110. La synchronisation par retour en arrière lors de la détection d'un évènement ...	147
Figure 111. L'interface de co-simulation de l'ICAr.	148
Figure 112. Principe de la procédure de mise en place du couplage	150
Figure 113. Les résultats de la co-simulation entre COMFIE et Brahms dans Simulink	152
Figure 116. Le plan de chauffage et ventilation du 1 ^{er} étage de Predis	185
Figure 117. Modélisation acausale de la partie électrique de la VMC dans Modelica	188
Figure 118 : Tableau des variables des formules de calcul de l'efficacité de l'échangeur thermique en fonction du régime d'écoulement [WET 1999a].....	194

Figure 119. Le plug-in Matlab/Simulink.....	195
Figure 120: Tableau des routines exécutées en fonction du flag envoyé [SIM 2010].....	196
Figure 121. Illustration de l'interaction entre le solveur de Matlab/la S-fonction et l'ICAr .	197
Figure 122. L'interfaçage ICAr / C++	198
Figure 123. Exemple d'appel de fonction externe C dans Modelica	199
Figure 125. Le principe du plug-out d'un modèle de COMFIE.....	200
Figure 126. Implémentation de l'interface ODE de l'ICAr COMFIE	201
Figure 127. Le prototype d'un nouveau « TYPE » de TRNSys	202
Figure 128. Tableau des informations communiquées par les éléments du vecteur « info »	202
Figure 129. Tableau des informations communiquées par la valeur de l'élément 7 du vecteur « info »	203
Figure 130. Le principe du plug-in TRNSYS	203

Table des tableaux

Tableau 1. Les natures des modèles supportés dans les outils utilisés.....	38
Tableau 2. Les capacités d'interopérabilités des outils utilisés.....	42
Tableau 3. Les valeurs des paramètres identifiées du circuit équivalent	62
Tableau 4. Valeur du rendement selon le type de ventilateur	74
Tableau 5. Le scénario d'activité décrit dans Brahms	78
Tableau 6. Les variables efforts flux de quelques domaines physiques dans Modelica	88

Table des annexes

Annexe 1: Structure de G-HomeTech.....	183
Annexe 2: Présentation générale de TRNSYS	184
Annexe 3: Le plan du 1 ^{er} étage de Predis	185
Annexe 4: Principales étapes de la construction du modèle de l'enveloppe thermique par COMFIE.....	186
Annexe 5: Le modèle de la partie électrique de la VMC (moteur et variateur de vitesse) dans Modelica.....	188
Annexe 6: Tableau des formules de calcul de l'efficacité de l'échangeur thermique en fonction du régime d'écoulement.....	194
Annexe 7: Le plug-in Simulink.....	195
Annexe 8: Le plug-in DYMOLA	198
Annexe 9: Le plug-out COMFIE	200
Annexe 10: Le plug-in TRNSYS	202

Acronymes

ANR: Agence Nationale de Recherche
API: Application Programming Interface
AUTOSAR: AUTomotive Open System Architecture
BBC: Bâtiments Basses consommations
BCVTB: Building Controls Virtual Test Bed
BDI: Belief, Desire, Intention
BEPos: Bâtiments à Energie Positive
BI: Bâtiments Intelligents
BPS: Building Performance Simulation
BVM: Brahms Virtual Machine
DAE: Differential Algebraic Equation
DF: Double flux
DLL: Dynamik Link Library
DPE: diagnostique des Performances Energétiques
DYMOLA : Dynamic Modeling Laboratory
EOO: Equation Oriented Object language
EPEX: European Power Exchange
FMI: Functional Mockup Interface
GES: gaz à effet de serre
GIEC: Groupe d'experts Intergouvernementaux sur l'Evolution du Climat
kWh_{EP}: Kilo wattheures exprimée en énergie primaire (EP)
GTC: Gestion Technique Centralisée
GTB: Gestion Technique du Bâtiment
HQE: Haute Qualité Environnementale
HVAC: heating Ventilation Air Conditioning
ICAr: Interface Component Architecture
IDM: Ingénierie Dirigée par les Modèles
IEEE: Institute of Electrical and Electronics Engineers
IFC: Industry Foundation Classes
IGBT: Insulated Gate Bipolar Transistor
IHM: Interface Homme Machine
IS: Ingénierie Système
JAR: Java Archive, développé par *Sun Microsystems*
JDK: Java Development Kit

JNI: Java Native Interface

JVM: Java Virtual Machine

MAS : Machine Asynchrone

MDE: Maîtrise de la Demande Energétique

MDE: Model-Driven Engineering

MHI: Monitoring Habitat Intelligent

MLI: Modulation de Largeur d'Impulsion

MTep: Tep: Tonne d'Equivalent Pétrole est une unité internationale de mesure de l'énergie. Le Mtep est un MégaTonne d'Equivalent Pétrole. (1 MTep = 42 PJ = $42 \cdot 10^{15}$ J, 1 TWh = 10^9 kWh = 3,6 PJ).

MUSE: Modèle Unifié pour les Systèmes Energétique des bâtiments

NMF: Neutral Model Format

ODE: Ordinary Differential Equation

OSGI: Open Services Gateway initiative

PAC: Pompe A Chaleur

PDE, EDP: Partial Derivative Equation, Equations aux dérivées partielles

PLUMES: Plateforme Logicielle Unifiée de Modélisation pour l'Efficacité énergétique du bâtiment et de ses Systèmes

RTE: Réseau de Transport d'Electricité

RT: Réglementation

SGEB: Systèmes de Gestion Energétique du Bâtiment

SIMINTHEC: SIMulation et outils logiciels INteropérables pour la pour la gestion des énergies THERmiques et Electrique dans les bâtiments

TRNSYS: TRaNsient SYstem Simulation Program

VHDL_AMS: Very high speed integrated circuits Hardware Description Language
– Analog and Mixed Signal extensions

VM: Virtual Machine

VMC: Ventilation Mécanique Contrôlée

UHT: Usage Habitat Tertiaire

Nomenclatures

Modèle COMFIE	
T	température
U	vecteur des sollicitations
Y	vecteur des sorties
X	vecteur d'état
\dot{X}	dérivé du vecteur d'état
A, E, J, C, G, F, H, B ^α , B ^β , S ^α , S ^β	Matrices d'états
Modèle réduit de l'enveloppe	
T _{mur}	température à l'intérieur des murs
T _{int}	température ambiante de la zone étudiée
T _{ext}	température extérieure
T _{zone}	température ambiante de la zone voisine
C _m	capacité des murs
R _m	résistance interne des murs
R _f	résistance de filtrage
R _{vide}	résistance des murs avoisinants le vide
R _{eprech}	résistance du sol séparant avec l'espace EP recherche
R _{couloir}	résistance des murs avoisinants les couloirs
R _{shed}	résistance des murs avoisinants le shed
R _{bureau}	résistance de la paroi avoisinant l'espace bureaux
Φ _{chauf}	flux thermiques dues aux systèmes de chauffage
Φ _{usage}	flux thermiques dues aux usagers
Φ _{elec}	flux thermiques dues aux systèmes électriques
Φ _{solaire}	flux solaires
Moteur et variateur de vitesse	
P _n	puissance nominale
Ω _n	vitesse de rotation nominale
η _n	facteur de puissance
V _m	tension simple appliquée au moteur
f _m	fréquence appliquée au moteur
cosΦ _m	facteur de puissance du moteur
I _m	courant entrée statorique du moteur
V _b	tension du réseau du bâtiment
f _b	fréquence du réseau du bâtiment
Ω _{consig}	consigne de vitesse de rotation
Ω _r	vitesse de rotation de l'axe du moteur
C _{ré}	couple résistant
η _m	rendement du moteur
η _{ep}	rendement du circuit de EP du variateur de vitesse
I _s	courant statorique
R _s	résistance statorique
V _s	tension au borne d'un enroulement statorique
I _r	courant rotorique ramenée au stator
R _r	résistance équivalente du rotor ramenée au stator
X	réactance équivalente ramenée au stator

g	glissement
I_{μ}	courant magnétisant
R_{μ}	résistance des pertes de fer
X_{μ}	réactance magnétisante
f_h	fréquence de hachage
V_{CE0}	tension de seuil de l'IGBT
$I_{IGBTmoy}$	courant passant moyen
R_{IGBT}	résistance passante de l'IGBT
$I_{IGBTeff}$	courant passant efficace
V_{d0}	tension de seuil de la diode
I_{dmoy}	courant passant moyen
R_d	résistance passante de la diode
I_{deff}	courant passant efficace
W_{on}, W_{off}	énergie de commutation à l'ouverture et à la fermeture
$W_{on\ ref}, W_{off\ ref}$	énergie de commutation de référence à l'ouverture et à la fermeture
E_{ref}, I_{ref}	tension continue et courant passant de référence
E, I	tension continue et courant moyen d'une période de commutation
I_d	courant passant de la diode
$I_{d\ ref}$	courant passant référence
$Q_{rr\ ref}$	charge totale recouverte de référence
E	tension continue contrainte
Echangeurs de chaleur	
i	indice du fluide (1 entrant et 2 sortant, a pour air et e pour eau)
g	grandeur caractéristique du fluide : T (température), D, C, Q...
g^{nom}	<i>grandeur nominale</i>
g_{ar}	<i>grandeur caractéristique de l'air rejeté</i>
g_{an}	<i>grandeur caractéristique de l'air neuf</i>
g_{ie}	<i>grandeur du fluide i à l'entrée de l'échangeur</i>
g_{is}	<i>grandeur du fluide i à la sortie de l'échangeur</i>
g_{cc}	<i>grandeur d'un échangeur statique à contre courant</i>
D	débit volumique (m ³ /h)
Q	puissance thermique (W)
C_i	capacité calorifique du fluide i (J/°K)
C_{min}	capacité calorifique minimale entre celle des deux fluides
C_{max}	capacité calorifique maximale entre celle des deux fluides
c_p	chaleur spécifique (J/kg·°K)
\dot{m}	débit massique (kg/s)
UA	conductance thermique= coefficient de transfert de chaleur* la surface d'échange (W/°K)
hA	coefficient de transfert convectif * la surface d'échange (W/°K)
NUT	nombre d'unités de transfert
ε	efficacité
M	masse de la roue accumulatrice (kg)
N	vitesse de rotation de la roue (tr/s)
c_m	chaleur spécifique de la masse de la roue (aluminium) (J/kg·°K)
Circuit aéraulique	
Δp_i	différence de pression dans le composant i du circuit

Δp_s	augmentation de pression subie par l'air
Δp_{th2}	augmentation de pression subie par l'air en absence de pertes aérauliques
q_v	débit volumique
η_{air}	rendement aéraulique du ventilateur
η_{vol}	rendement volumétrique du ventilateur
η_s	rendement global du ventilateur
$\eta_{méc}$	rendement mécanique du ventilateur

Introduction générale

Dans un nouveau contexte énergétique mondial caractérisé par des fortes contraintes de réduction des émissions en gaz à effet de serre, des résolutions importantes ont été prises dans les différents secteurs énergivores. Le secteur du bâtiment, premier consommateur mondial d'énergie, représente une charge importante et, par conséquence, une variable d'ajustement à ne pas négliger. Pour mieux maîtriser ses flux énergétiques et respecter les diverses restrictions mises en place, le bâtiment devient de plus en plus complexe. Il par ailleurs soumis à des variations de facteurs extérieurs et internes qu'il est nécessaire de considérer.

Cette complexité « système » demande un changement dans les habitudes de simulation du bâtiment pour la prise en compte de ces évolutions. Une modélisation globale des différents composants de ce système et une simulation efficace des ses sous-systèmes hétérogènes, doivent dorénavant être assurées. Ces objectifs ne pourront être atteints qu'au prix d'un changement de paradigme dans les applications de conception, de simulation et de supervision. Ceci exige en effet d'aller au delà du paradigme actuel d'environnements, d'outils de simulations et des systèmes de gestion technique non inter-opérables, vers un nouveau paradigme orienté « modèle » pour représenter les connaissances des différentes composantes du système. Il va ainsi s'appuyer sur des approches méthodologiques, concepts informatiques et langages de description pour créer, transformer et utiliser facilement des modèles, qui seront conceptuellement dédiés à l'interopérabilité. Nous allons ainsi montrer l'intérêt de ce transfert de paradigme à travers deux approches complémentaires d'interopérabilité : l'approche boîte blanche se basant sur le langage Modelica et l'approche boîte noire s'appuyant sur le concept de composants logiciels.

Dans le premier chapitre de ce manuscrit, nous commençons par exposer le contexte énergétique mondial et français en relation avec le secteur du bâtiment pour conclure sur la nécessité d'une modélisation et d'une simulation globale. Cette dernière se trouve confrontée à des problèmes relatifs à l'hétérogénéité des modèles et à la limitation des simulateurs actuels. Un état de l'art des solutions existantes d'interopérabilité dans la filière sera élaboré dans l'objectif de dépasser ces limites et pour conclure sur le manque de standardisation et le besoin de mettre en place des approches plus avancées et plus génériques d'interopérabilité.

Le deuxième chapitre illustrera, en application au système de bâtiment basse consommation PREDIS (installé dans les locaux du G2ELAB), la diversité des natures et des typologies des modèles rencontrés dans le bâtiment. Les difficultés relatives à l'élaboration de

chacun de ces modèles seront mises en évidence pour conclure sur la nécessité de favoriser la réutilisation de modèles existants. Cette technique de réutilisation rend la modélisation globale d'un système de bâtiment plus délicate nécessitant des solutions efficaces d'interopérabilité.

Dans le troisième chapitre, seront étudiées les deux approches d'interopérabilité proposées dans cette thèse pour palier aux difficultés de modélisation système rencontrées et gérer l'hétérogénéité des divers modèles utilisés. Les intérêts et les limites de la première approche se basant sur le langage Modelica seront ainsi mis en évidence en application au bâtiment PREDIS. Les avantages de l'introduction d'un standard de composant logiciel adapté à ce secteur, en complément de l'approche Modelica, seront montrés à travers la mise en place d'un bus à composant logiciel. Ce bus est enrichi par divers développements depuis et vers des logiciels de simulation spécifiques de certains domaines du bâtiment (thermique, électrique, modélisation du comportement des usagers...).

Le quatrième chapitre se focalise sur les aspects de simulation des sous systèmes hétérogènes (modèles et/ou outils) et les moyens permettant d'assurer une simulation performante tenant compte de leurs spécificités. Cette étude sera établie dans l'objectif de définir les spécifications d'une norme de composant logiciel susceptible d'être générique et adaptée à divers typologies de modèles et de stratégies d'orchestration. L'introduction de cette approche dans le processus de modélisation et simulation sera également présentée à travers des cas d'application tirés de PREDIS.

Chapitre 1:

Contexte énergétique, problématiques et état de l'art des solutions existantes d'interopérabilité de la simulation dynamique du bâtiment

I. Introduction

Par rapport au défi de réduction des émissions de CO₂ dans le bâtiment, premier secteur consommateur d'énergie, il est indispensable de limiter sa consommation énergétique en améliorant ses performances à travers une meilleure isolation thermique et des ponts thermiques moins importants, en y intégrant des énergies renouvelables mais aussi en assurant une gestion optimale de sa consommation qui doit prendre en compte tous les acteurs. En effet, dans le cadre d'un nouveau contexte de bâtiments basses consommations qui sont bien isolés grâce à des matériaux et des techniques très avancées, les apports internes (usagers, appareils électriques) deviennent de plus en plus influents et ne peuvent plus être négligés [CHEN 2010]. Ainsi une étude fiable du bâtiment doit pouvoir représenter ses différents acteurs issus de plusieurs domaines et les coordonner de manière à tenir compte de leurs différences et interactions au sein de ce système.

Or les avancées de la simulation dynamique de chaque domaine dans le bâtiment (thermique, électrique, éclairage, aérodynamique, contrôle) sont considérables, donnant naissance à des logiciels pertinents dans le domaine qu'ils traitent et dotés souvent de bibliothèques métiers si riches que leurs re-développements dans d'autres environnements de simulation paraît inconcevable. Ainsi le champ d'application de chaque outil de simulation reste restreint au domaine qu'il cible.

A nos jours, le besoin d'utilisation des outils métiers permettant de dimensionner et de valider les systèmes de gestion des flux énergétiques dans le bâtiment en y intégrant tous les aspects allant des énergies renouvelables aux systèmes de contrôle en passant par les aspects électriques et thermiques mais encore par la simulation du comportement des habitants, a nécessité le recours et le développement de diverses solutions d'interopérabilité. Cette interopérabilité peut être réalisée au niveau des données, des modèles ou des simulateurs pour pouvoir réaliser des simulations complètes du système global.

Ces différents aspects seront abordés tout au long de ce chapitre pour conclure sur le manque de standardisation dans le domaine du bâtiment et la nécessité d'introduire un nouveau standard dédié à l'interopérabilité.

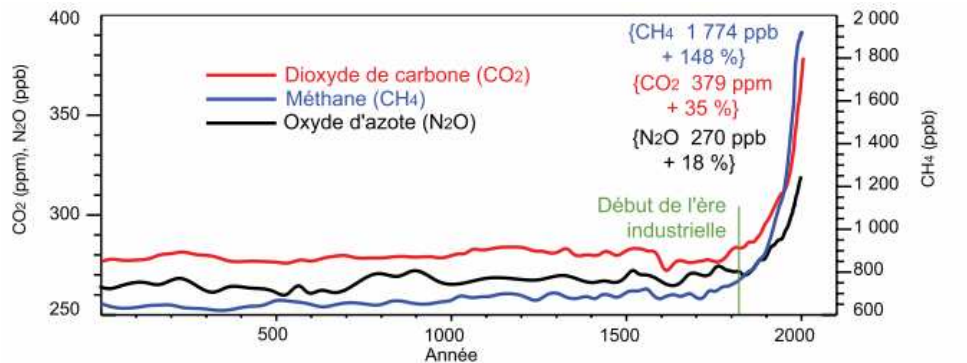
II. Contexte énergétique et environnemental

II.1. Le bâtiment : un enjeu énergétique majeur

II.1.1. Le bâtiment : un secteur polluant et consommateur d'énergie

II.1.1.1. Au niveau mondial

Ces dernières années ont été marquées par une évolution rapide et alarmante de la concentration du gaz à effet de serre (GES) (Figure 1) et essentiellement de l'augmentation du taux de CO₂ dans l'atmosphère. Il est de plus en plus probable que ces taux élevés pourraient induire un réchauffement climatique considérable qui pourrait avoir un fort impact sur la vie humaine. Le réchauffement climatique risque d'induire des conséquences considérables et un changement radical de la configuration actuelle de la planète (remontée du niveau d'eau des océans, la dégradation de la biodiversité...).



Les chiffres entre crochets correspondent à la concentration atmosphérique du gaz en 2005 et à son taux de croissance entre 1750 et 2005.

Source : GIEC, 1^{er} groupe de travail, 2007.

Figure 1. Evolution du taux de concentration de différents gaz à effet de serre

Cette croissance des taux de pollution semble être en parfaite corrélation avec l'évolution exponentielle de la consommation mondiale d'énergie primaire depuis 1971 (Figure 2). Elle atteint en 2009 une valeur de 12150 Mtep (soit environ 141.10^{12} kWh) [IEA 2011].

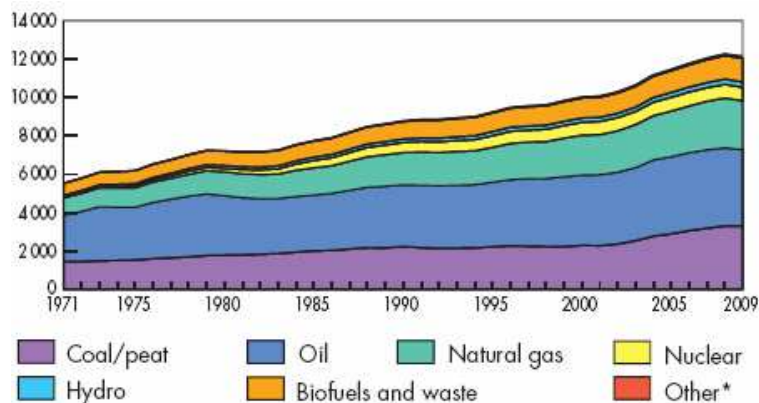


Figure 2. Evolution de la consommation d'énergie primaire mondiale (en Mtep) selon la source [IEA 2011].

Cette consommation constamment croissante conduit à l'épuisement accéléré des ressources naturelles mais aussi à des taux anormalement élevés de pollution. L'homme se

trouve ainsi confronté à des inquiétants défis environnementaux auxquels il doit faire face et prendre des mesures efficaces. Dans cet objectif, le protocole de Kyoto [KYO 1998] a été établi engageant plusieurs nations à réduire leurs émissions de gaz à effet de serre entre 2008 et 2012, de 5,2% par rapport au niveau de 1990. Pour atteindre ces objectifs, il est indispensable d'agir sur les secteurs les plus émetteurs de CO₂.

Le secteur du bâtiment (résidentiel et tertiaire), est un gros consommateur d'énergie au niveau mondial avec une part de 34% de la consommation globale de l'énergie finale en 2004 [EDD 2004]. Il se positionne donc comme un acteur clé pour parvenir à résoudre les défis environnementaux auxquels les pays doivent faire face et respecter leurs engagements.

II.1.1.2. Au niveau de la France

Au niveau de la France, les mêmes tendances peuvent être retrouvées avec un secteur bâtiment qui présente la partie majoritaire des consommations énergétiques annuelles de 43% de l'énergie finale en 2010 (Figure 3) provoquant ainsi une émission de CO₂ de 21% [CITEPA 2010].

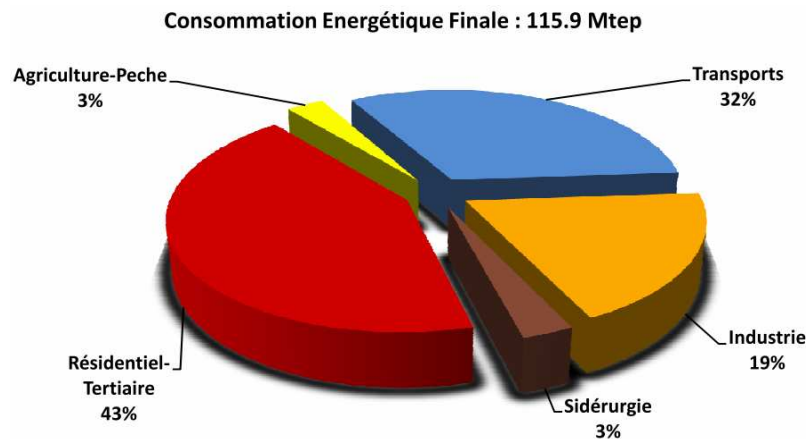


Figure 3. Répartition de la consommation d'énergie par secteur en France [SOES 2010]

Au cours de ces trente dernières années, la consommation énergétique des bâtiments a connu une remarquable augmentation en passant de 47,4 Mtep en 1970 à 69,4 Mtep en 2008 [ADEME 2009] du fait de l'accroissement du nombre des logements (+ 41% de logements en 30 ans [ADEME 2005]) ainsi que de leurs surfaces moyennes et de l'augmentation des besoins en terme de confort et de nouvelles technologies. Une baisse de cette consommation a été notée en 2009 (-0,9% [ADEME 2010]) et en 2010 (- 1,2% [ADEME 2011]), considérées atypiques, à cause des effets de la crise économique.

Le secteur du bâtiment pourrait bien être le seul qui offre des possibilités de progrès important permettant de répondre aux engagements nationaux et internationaux. Le processus de concertation du Grenelle de l'Environnement [MEDAD 2006] a proposé des objectifs ambitieux de réduction des émissions de gaz à effet de serre qui doivent être divisés par quatre à l'horizon de 2050, conformément aux recommandations du GIEC. Le plan bâtiment à travers la réglementation thermique [RT 2012] exige la norme de 50 kWh_{EP}/m²/an alors que la consommation moyenne annuelle est de 240 kWh_{EP} par m² chauffé en 2008 (tous bâtiments confondus : anciens et neufs) [ADEME 2008]. Donc, malgré les efforts réalisés par la France depuis son engagement dans le protocole de Kyoto (Figure 4), un travail considérable reste

encore à faire pour améliorer les performances énergétiques des bâtiments actuels et des bâtiments futurs.

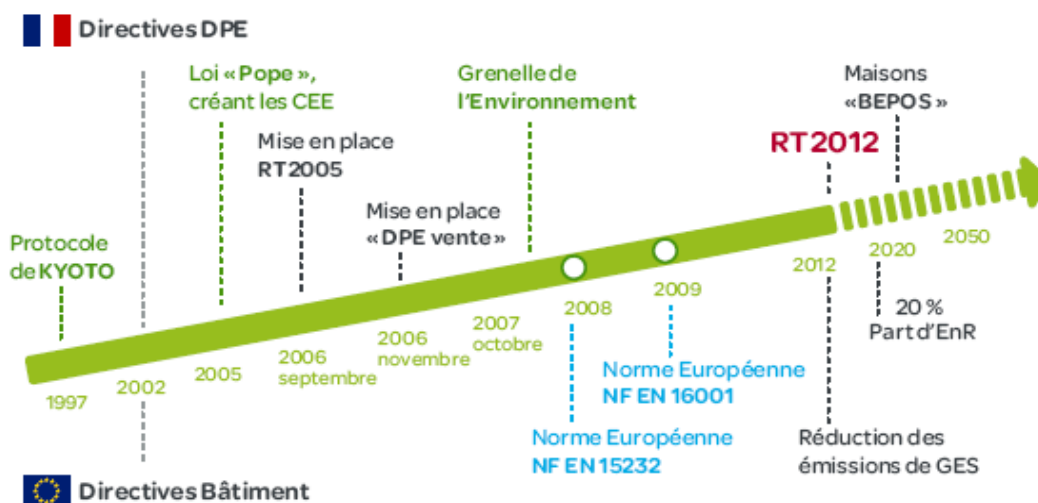


Figure 4. Les efforts réalisés par la France depuis 1997 pour améliorer l'efficacité énergétique des bâtiments (directives DPE) [ADEME 2010]

II.1.2. Le bâtiment : une charge électrique importante

II.1.2.1. Le bâtiment est le premier secteur consommateur d'énergie électrique

Le secteur du bâtiment, avec ces deux composantes résidentielle et tertiaire, est de loin le premier poste de consommation électrique en France avec 65 % de la consommation électrique globale de 442 TWh (Figure 5). Cette consommation est en constante augmentation depuis 1970 (Figure 6).

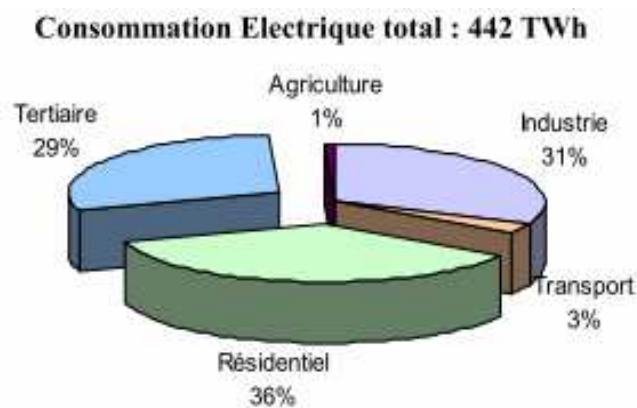


Figure 5. Répartition de la consommation électrique en France par secteur en 2010 [CGDD 2010]

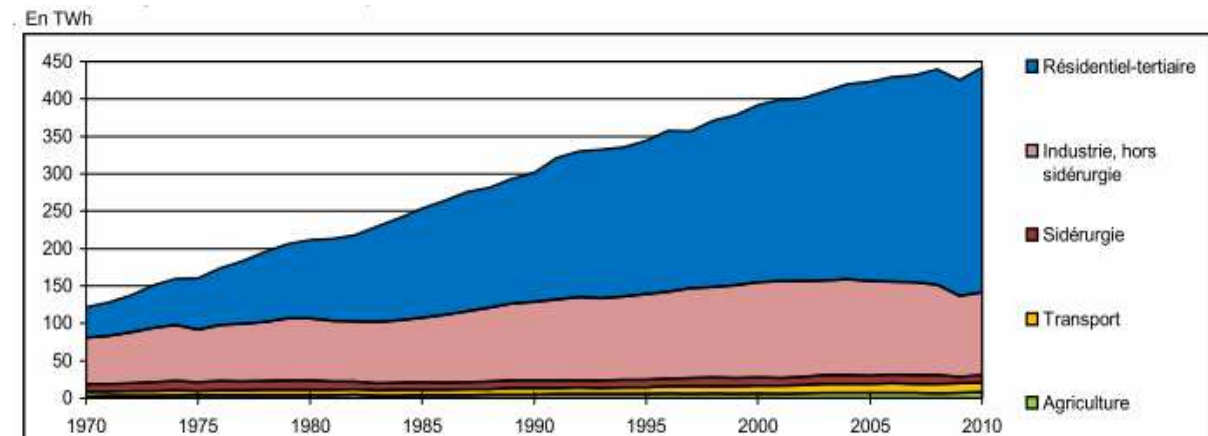


Figure 6. L'évolution de la consommation électrique en France corrigée des variations climatiques par secteur [CGDD 2010]

II.1.2.2. L'électricité est le premier vecteur d'énergie consommé dans le secteur du bâtiment

L'électricité est classée comme la première énergie consommée dans le secteur résidentiel-tertiaire en France avec 38% de l'énergie globale consommée des 68,1 Mtep en 2010 (Figure 7) et une croissance de 4,3% [ADEME 2011].

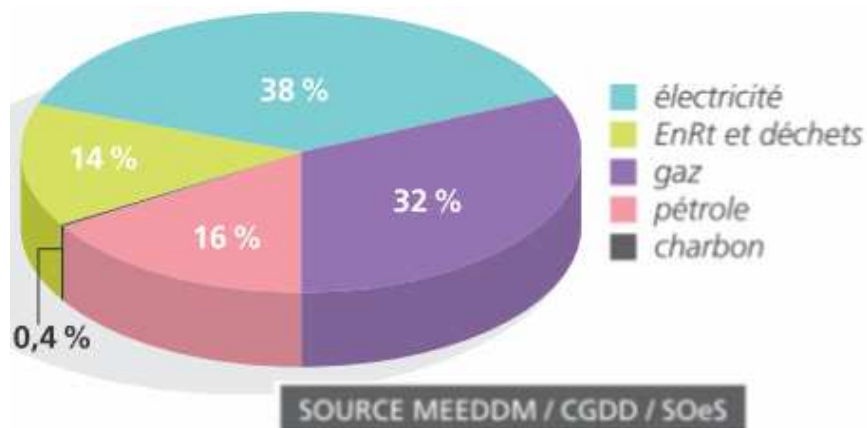


Figure 7. Répartition des sources d'énergie consommée dans le secteur résidentiel-tertiaire français en 2010 [ADEME 2011]

Cette part importante est due principalement à:

- La hausse de la part du chauffage électrique dans les maisons résidentielles (essentiellement dans les maisons neuves car moins cher à l'installation) qui a passé de 32,5% en 2009 [ADEME 2010] à 33% en 2010 [ADEME 2011]. Cette hausse est observée malgré une tendance globale de la baisse des consommations dues au chauffage (baisse de 55% depuis 1973 [ADEME 2010]) grâce à l'amélioration de l'isolation des bâtiments et la maîtrise des déperditions thermiques. Mais le chauffage reste le premier poste de consommation de l'énergie avec 62,7% des consommations des résidences principales [ADEME 2011].
- L'installation de nouveaux systèmes de chauffages dans le secteur tertiaire (pompe à chaleur, VMC, ...) nécessitant l'utilisation de l'énergie électrique.

- La multiplication des « appareillages électriques spécifiques¹ ». La consommation unitaire moyenne d'électricité spécifique est passée de 13 kWh/m² en 1973 à 30 kWh/m² en 2008 [ADEME 2010] (Figure 8). Ce phénomène s'explique notamment par la progression de l'équipement en appareils électroménagers, hi-fi et bureautique.

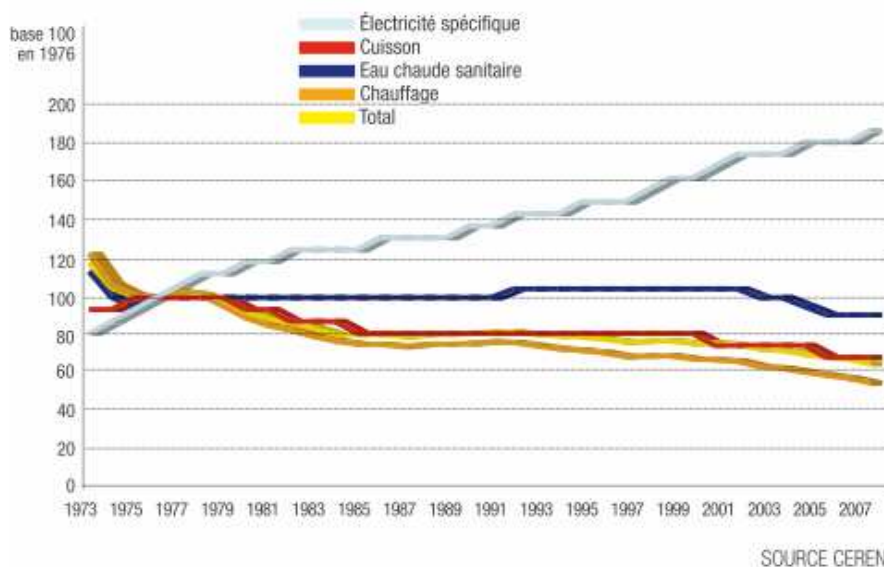


Figure 8. Evolution des consommations unitaires des résidences principales par usage [ADEME 2010]

II.1.2.3. Conséquences de cette charge importante

Cette consommation importante de l'énergie électrique entraîne non seulement des émissions importantes de CO₂, des coûts élevés, mais aussi des sources thermiques non négligeables causant l'élévation des températures essentiellement dans des bâtiments thermiquement isolés [CHEN 2012]. Cet effet peut être exploité en hiver pour aider au chauffage du bâtiment, bien qu'il nécessite une gestion plus complexe du fait de la non maîtrise de ses sources. Mais cet effet nuit principalement au « confort d'été » venant accroître la consommation des systèmes de rafraîchissement.

Les acteurs électriques deviennent ainsi indispensables pour assurer le bon choix de la structure architecturale ainsi qu'au bon dimensionnement et fonctionnement des équipements dès la phase de conception.

L'utilisation de matériels performants tels que des éclairages LFC (lampe fluorescente compacte) ou à LED, des alimentations à découpage au lieu de transformateurs... peut nettement réduire la consommation d'électricité. Mais malgré leurs faibles puissances, ces appareils peuvent, quand même, consommer une quantité conséquente d'électricité du fait des mauvais usages tel que l'éclairage inutile, les mises en veilles plutôt que l'extinction totale [POQU 2008].

Outre la quantité d'électricité consommée, la période de consommation demeure également importante. Il serait judicieux, par exemple, de programmer la charge des batteries

¹ Appareils électriques spécifiques: autres que chauffage/rafraîchissement, eau chaude sanitaire, éclairage et auxiliaires (ventilation, pompe).

(ordinateurs portables, véhicules électriques...) pendant des heures creuses ... Donc la bonne gestion des ces équipements s'avèrent indispensable pour une meilleur maîtrise de leurs consommations énergétiques.

II.1.3. Le bâtiment : une variable d'ajustement du réseau électrique

II.1.3.1. Le bâtiment est un acteur du réseau électrique

L'énergie électrique est acheminée depuis son lieu de production (essentiellement des centrales nucléaires en France avec 63,1 GW en 2010 ce qui correspond à 80% de la production globale) jusqu'aux postes de consommations à travers le réseau électrique : réseau de transport haute tension et réseau de distribution moyenne et basse tension. Ainsi, au dernier nœud de distribution, l'énergie est acheminée aux différents utilisateurs : le secteur résidentiel- tertiaire, le secteur industriel, et le secteur du transport avec une part respective de 64%, 30% et 3% (Figure 9).

Le bâtiment est donc un acteur majeur du réseau électrique qui peut jouer un rôle primordial dans la résolution des problèmes liés à l'exploitation des réseaux électriques comme les pics de consommations et les black-outs.

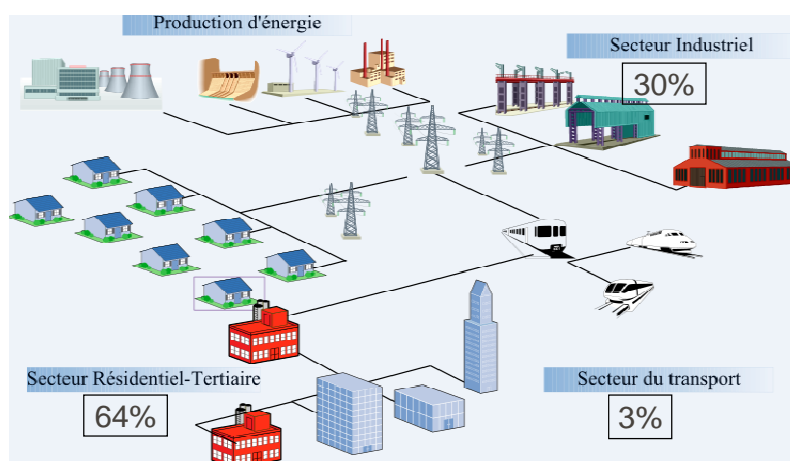


Figure 9. Les acteurs du réseau électrique Français

II.1.3.2. Lisser les courbes de consommations et réduire les pics de consommations

Un pic de consommation est un record de consommation d'électricité. Il s'agit d'un dépassement des seuils d'appels de courant pour lesquels les installations électriques du réseau ont été dimensionnées. La répétition fréquente de ce phénomène, lié à une plus ou moins longue durée, peut entraîner l'effondrement d'une partie du réseau électrique.

Ce phénomène est la conséquence des conditions météorologique extrêmes (grand froid : chauffage ou grande chaleur : climatisation). Pour la France, les pics ont généralement eu lieu en hiver lors des saisons où les températures sont inférieures à 5°C et proches de 0°C en moyenne (Figure 10). En effet, la baisse d'un seul degré de température est compensée par une augmentation instantanée de 2300 MW sur toute la France [RTE].

La valeur des pics de consommations électriques en France suit une tendance croissante depuis une dizaine d'année (Figure 10). Cette croissance est directement liée à la multiplication des équipements électriques des bâtiments, dont le chauffage électrique. Au

cours de l'année 2012, un nouveau pic de consommation électrique a été établi le 8 Février vers 19 heures à plus de 100 000 mégawatts suite à une vague de froid qui a envahi la France.

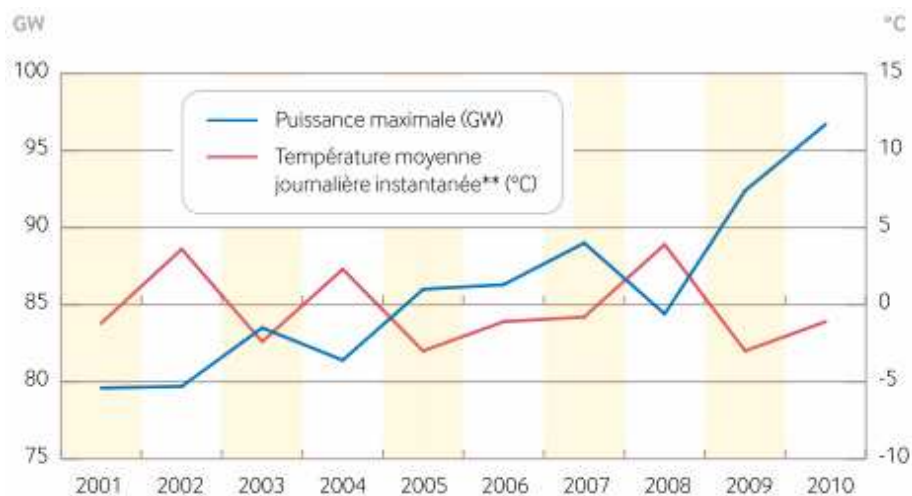


Figure 10. L'évolution des puissances électriques maximales consommées depuis 2001 en fonction des températures [RTE 2011]

La consommation électrique typique journalière varie d'une saison à une autre (Figure 11). Le pic de consommation journalier hivernal est détecté typiquement vers 19h ce qui correspond au retour du travail et la mise en route des équipements dont essentiellement le chauffage. Ce pic est également détecté en été, à la même période de la journée même si cela est avec une valeur plus faible de la consommation, et un autre pic est atteint vers 13h.

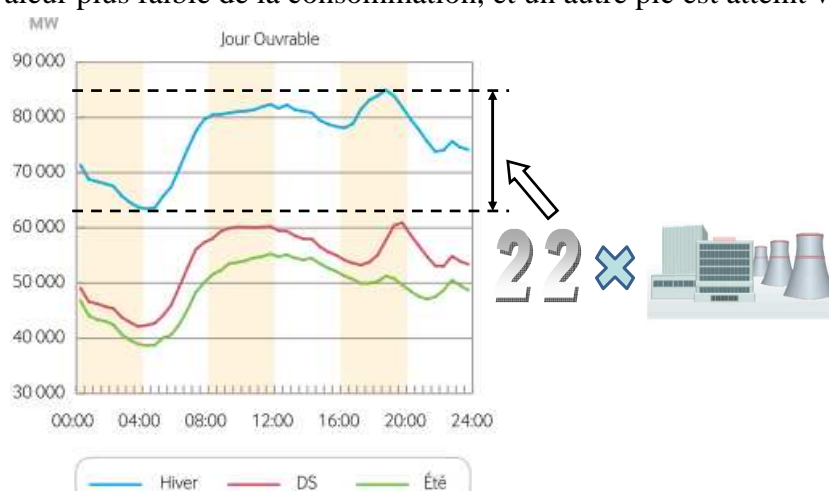


Figure 11. Consommations typiques journalières en été, en demi-saison (DS) et en hiver [RTE 2011]

La variation entre le minimum et le maximum de consommation dans une journée typique hivernale est de l'ordre de 22 GW, soit 22 tranches nucléaires (environ 1000 MW par tranche) (Figure 11). Ces pics sont donc très polluants et la réduction de leur fréquence et de leur puissance permettraient d'optimiser le dimensionnement de la production, en supprimant éventuellement des centrales inutiles et polluantes, ainsi que celui du transport et de la distribution.

Outre la baisse des consommations globales, la réduction des pics de consommation pourra passer par le lissage des courbes de consommations journalières à travers des actions de Maîtrise de la Demande Énergétique (MDE) comme le déplacement de charges vers les

heures de faible consommation [FOG 2009] [BOED 2009] [MIS 2012]. On peut par exemple prévoir de déplacer les charges de 19h vers 4h du matin. C'était l'objectif de la mise en place de la tarification heures creuses / pleines, pour mieux inciter les utilisateurs du réseau à déplacer leurs charges vers les heures creuses où le tarif de l'électricité est moins chers.

Mais cette action n'est plus suffisante, et il faudra aller vers une tarification dynamique de l'énergie (coût qui varie à chaque heure selon l'offre et la demande) [UFE 2010], comme c'est déjà le cas sur le marché EPEX [WEB_EPE]. Cette tarification dynamique pourra être répercutée à l'utilisateur final, grâce aux compteurs intelligents type linky [WEB_LIN] et à des nouveaux contextes juridiques, comme la loi NOME (nouvelle organisation du marché de l'électricité) [MAZ 2011].

II.1.3.3. Diminuer les « black-outs »

La consommation électrique en France représente ainsi des variations importantes le long d'une journée avec des pointes de consommation qu'il faut chercher à réduire. Pour satisfaire cette demande en électricité, RTE [WEB_RTE] vise à établir une prévision, la plus précise possible, de la consommation journalière afin de déterminer les ressources nécessaires à mettre en œuvre pour satisfaire la demande et établir un équilibre entre l'offre et la demande.

Un déséquilibre peut conduire à de sérieuses pannes électriques et une dérégulation du réseau électrique telle que la chute des niveaux des tensions et des fréquences devant être maintenus stables pour le bon fonctionnement des appareils et systèmes électriques. On peut assister à une cascade de problèmes allant jusqu'au plongement d'une région ou d'un pays entier dans un noir total et même d'un continent en cas d'interconnexion entre réseaux, ce qui est le cas du réseau européen. Il s'agit alors d'une coupure électrique à large échelle appelée « black-out ».

On peut ici rappeler l'incident de Novembre 2006 où 10 millions d'européens situés dans plusieurs pays (France, Allemagne, Italie) se sont trouvés privés d'électricité suite à une panne trouvant son origine en Allemagne après la mise hors service de deux lignes hautes tensions pour permettre le passage d'un bateau. Se trouvant alors en déséquilibre avec une consommation nettement supérieure à la production, et afin de préserver le fonctionnement du réseau et d'éviter une panne générale (black out), "des systèmes de sécurité automatiques coupent brutalement" [MEM 2004] l'alimentation en électricité des habitations. "C'est le seul moyen pour éviter un effondrement complet" du réseau, a expliqué un membre du directoire de RTE.

Ainsi, le risque de défaillance du réseau électrique dépend de la différence entre la production disponible et la puissance appelée. Ce risque de défaillance, en lien avec des pics de plus en plus importants, devient de plus en plus grand et avec des durées plus importantes pouvant dépasser les niveaux économiquement et socialement acceptables, fixés à moins de 3 heures par an. Les prévisions de RTE pour 2016 dépassent ce plafond avec une durée moyenne estimée à 8h50 (Figure 12).

	2013	2014	2015	2016
Énergie de défaillance en espérance (GWh)	0.2	0.8	2.8	27.4
Espérance de durée de défaillance	0h05	0h22	1h14	8h50
Puissance manquante	-	-	-	2.7 GW

Figure 12. Trajectoire probable des risques de défaillance [RTE 2011]

Pour éviter un « black-out » global, un délestage ponctuel peut être effectué afin de réduire les dégâts [LU 2009] (effacement des charges). Il est également possible de profiter des sources décentralisées de production de l'énergie pour renforcer les sources principales (centrales nucléaires, thermiques, hydrauliques...).

La décentralisation de la production énergétique au niveau des utilisateurs (en particulier au niveau des bâtiments) due à l'intégration des énergies renouvelables introduit des difficultés supplémentaires au niveau de la gestion de la production globale du réseau électrique [CAIRE 2004]. En contre partie, elle permet de disposer de degrés d'ajustement sur la consommation en plus de l'effacement des charges. Ceci en favorisant l'autoconsommation des bâtiments et l'augmentation de la part des énergies renouvelables jusqu'à assurer leur autonomie et leur autosuffisance [YAMA 2005].

II.1.4. Conclusion

Le bâtiment est ainsi un nœud énergétique majeur, dont une bonne maîtrise et gestion permettra de réduire les rejets en GES, de diminuer considérablement les consommations énergétiques, et d'améliorer la performance et la sûreté du réseau électrique...

Une synergie doit être créée autour de ce secteur énergivore avec des actions concrètes permettant de gérer au mieux cette variable d'ajustement sur tous les niveaux.

II.2. Des nouvelles actions et résolutions dans le secteur du bâtiment

II.2.1. Motivations des efforts réalisés dans le secteur du bâtiment

Une enquête réalisée par l'IPOS [WEB_IPS] a révélée deux principales motivations (Figure 13) pouvant inciter la population à réaliser des efforts pour améliorer leurs habitats :

- Réduire le montant de la facture énergétique qui représente la source principale de motivation.
- Améliorer le confort, dont le niveau d'exigence est toujours en augmentation.

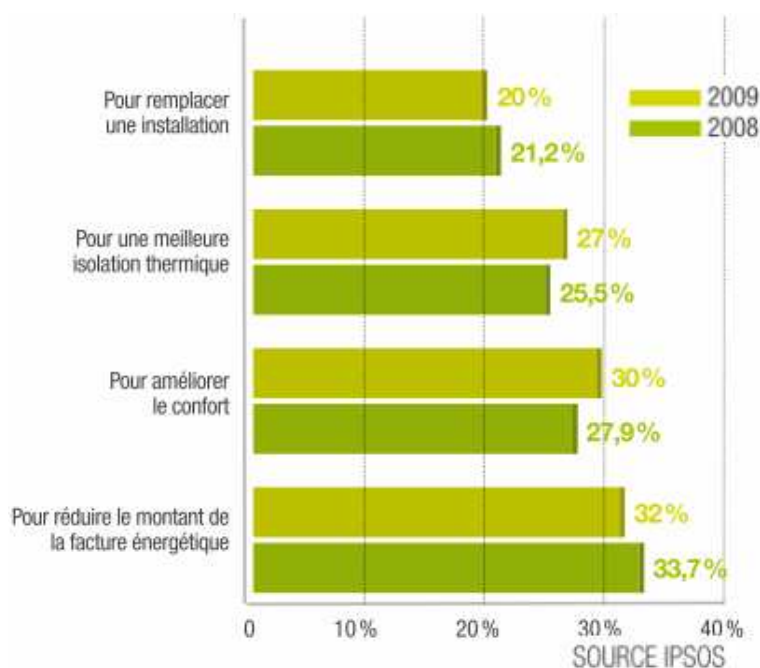


Figure 13. Les motivations pour l'amélioration de l'habitat [ADEME 2010]

Ces deux motivations et attentes de l'utilisateur doivent être prises en compte par les travaux de recherche, les bureaux d'études, et les pouvoirs publics dans ce secteur, afin que les solutions apportées soient largement appliquées sur le terrain. Les actions développées doivent assurer une réduction de la consommation énergétique, et ainsi de la facture, tout en respectant le confort des usagers. Ces deux axes sont d'ailleurs les axes principaux de la réglementation thermique 2005 [RT 2005] intitulée «des bâtiments confortables et performants ». Ces solutions s'organisent autour de deux axes principaux :

- Améliorer l'isolation thermique et favoriser l'autoconsommation à travers le concept des bâtiments à hautes performances énergétiques tels que les **bâtiments basse consommation**, voire des bâtiments à énergie positive.
- Maîtriser la demande énergétique et les usages pour assurer une « utilisation rationnelle de l'énergie » [CHAB 1997] qui respecte les préférences des occupants à travers la notion des **bâtiments intelligents** ou «smart buildings» [NIKO 2002].

II.2.2. Les bâtiments basses consommations, voire à énergie positive

II.2.2.1. Amélioration des performances énergétiques

Les bâtiments du futur devront être peu consommateurs d'énergie, voire autonomes énergétiquement. Pour atteindre ces objectifs et respecter ces engagements nationaux (le plan

climat), et internationaux [KYO 1998], la France a mis en place des normes sous forme de réglementations thermiques (RT) imposant l'utilisation d'une isolation thermique performante pour les maisons neuves ou en rénovation et qui respectent le Grenelle de l'environnement. Ces séries de réglementations thermiques (RT 1974, RT 1988, RT 2000, RT2005 [RT 2005], RT2012 [RT 2012]) sont de plus en plus exigeantes au fil du temps. Les exigences en terme de consommations énergétiques sont passées d'une moyenne de 150 kWh/ (m².an) dans la RT 2005 à 50 kWh/ (m².an) pour la RT 2012 (Figure 14).

Depuis la mise en place de la première RT en 74, les consommations énergétiques ont été divisées par 2 et avec la nouvelle RT de 2012, le Grenelle de l'environnement prévoit de la diviser encore par 3 pour pouvoir atteindre le plafond d'une moyenne de consommation de 50 kWh/ (m².an), conformément à l'article 4 de la loi Grenelle 1 [GRE 2009].

Il s'agit de la consommation moyenne des bâtiments basse consommation (BBC): le label de référence des maisons neuves à l'horizon de 2012, étape essentielle pour atteindre le niveau des bâtiments à énergie positive (BEPos) à l'horizon de 2020. Les bâtiments BEPos [THIE 2008] sont non seulement plus performants énergétiquement que le BBC, mais ils sont aussi caractérisés par l'intégration de production d'énergie renouvelable pouvant être injectée au réseau, rendant son bilan énergétique positif d'où son appellation.

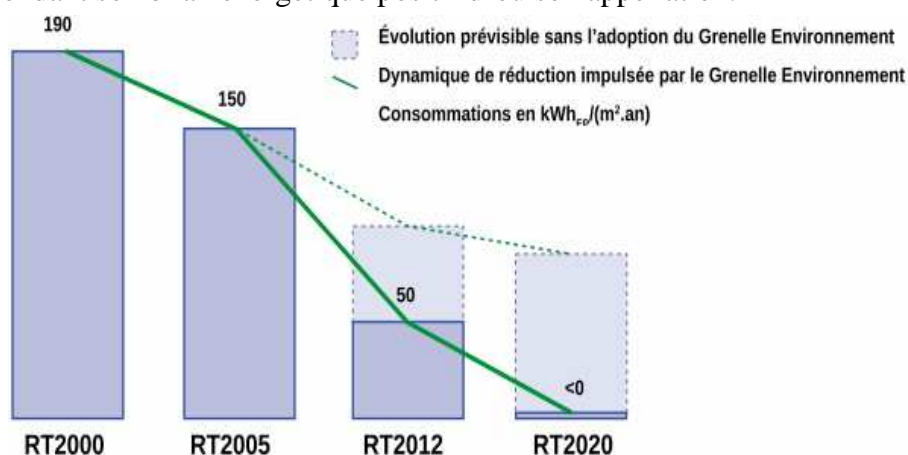


Figure 14. L'évolution des exigences réglementaires de la consommation énergétique des bâtiments neufs : une rupture opérée par le Grenelle Environnement [RT 2012]

Pour atteindre le niveau énergétique des BBC, la nouvelle RT 2012 [RT 2012] a imposé 3 principales exigences dans les maisons neuves :

- L'efficacité énergétique du bâtiment (Bbio): c'est un critère lié aux composantes de la conception du bâtiment (typologie, localisation, altitude...) sans tenir compte de ces systèmes énergétiques installés. Il s'agit ici de vérifier l'optimisation de l'inertie et de l'isolation du bâti pour réduire ses déperditions thermique en utilisant des matériaux performants (tels que la mousse de polyuréthane, les laine de verres ou le polystyrène) et aussi des techniques de mieux en mieux adaptées (telles que le double vitrage et la réduction des ponts thermiques). L'architecture de ces bâtiments est aussi étudiée de manière à assurer un apport passif maximal en hiver essentiellement des apports solaires à travers l'augmentation des surfaces vitrées et l'étude de leurs orientations.

- La consommation énergétique du bâtiment (Cep) : ce critère est lié à la consommation en énergie primaire des principaux usages : chauffage, refroidissement, éclairage, production d'eau chaude sanitaire et auxiliaires (pompes, ventilateurs...). Ceci exige l'intégration des systèmes énergétiques performants tels que la ventilation double flux, chaudière à condensation, chauffage thermodynamique (pompe à chaleur)...Et aussi l'utilisation des énergies renouvelables comme les panneaux solaires photovoltaïques [MANZ 2009].

- Le confort d'été (Tic : température intérieure conventionnelle) : il s'agit de vérifier l'aptitude du bâtiment à garantir un bon niveau de confort et de température en été sans recourir à des systèmes de refroidissement introduisant des consommations supplémentaires. Ce niveau de confort peut être assuré à travers la protection des locaux du soleil l'été (masques végétaux et protections solaires amovibles), un rafraîchissement optimisé (ventilation nocturne, brassage d'air), un décalage des pics de température (inertie du bâtiment), et aussi une prise en compte de la contribution des apports internes provoquant une élévation des températures que l'on essaiera de minimiser.

II.2.2.2. Conséquences de ces évolutions

Les bâtiments performant thermiquement ne fonctionnent pas comme des bâtiments classiques ; les acteurs internes et l'usage d'équipements deviennent de plus en plus influents sur la consommation énergétique et les phénomènes auparavant négligeables dans les bâtiments à 150 kWh/m²/an de besoins de chauffage deviennent de première importance dans les BBC (<50 kWh/m².an).

De plus, un bâtiment bien isolé thermiquement est davantage exposé aux surchauffes dues aux charges internes par un effet « Thermos » [AMO 2009]. En effet, une isolation importante empêche la chaleur de s'échapper et en préserve une grande partie à l'intérieur du bâtiment (Figure 15). Les apports solaires ainsi que les apports internes dégagés par les occupants et les équipements deviennent importants et participent considérablement dans le réchauffement de la pièce.

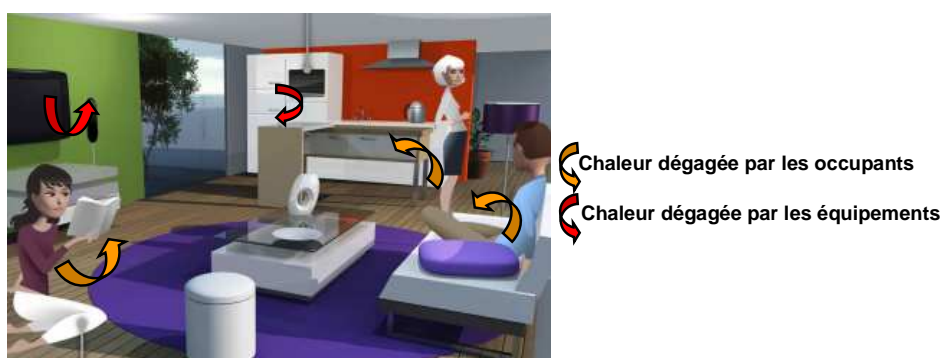


Figure 15. Illustration des apports internes

Cet effet thermos, peut entraîner des surchauffes estivales importantes rendant le confort d'été plus délicat. Beaucoup de bâtiments basse consommation sont victimes de cette surchauffe car si on pense intuitivement à protéger les locaux du soleil l'été, on sous-estime la contribution des apports internes à l'élévation des températures.

On peut ici citer comme exemple les études menées sur le bâtiment PREDIS pour quantifier ces apports internes. Le bâtiment PREDIS, qui sera étudié avec plus de détails dans le chapitre 2 de cette thèse, est un bâtiment basse consommation qui a été restructuré à partir d'un bâtiment existant en respectant la réglementation thermique (RT 2005). Une isolation performante à l'aide de ouate de cellulose ainsi qu'un système énergétique efficace (VMC double flux) de manière à garantir une consommation en chauffage inférieure à 50kWhEP/m²/an.

Les études réalisées par Hervé Chenailler [CHEN 2011] sur ce bâtiment ont évalué à 50 % la contribution des apports internes dans le chauffage annuel (Figure 16). L'importance des ces apports a même provoqués des périodes de surchauffes fréquentes en été en l'absence de système de climatisation.

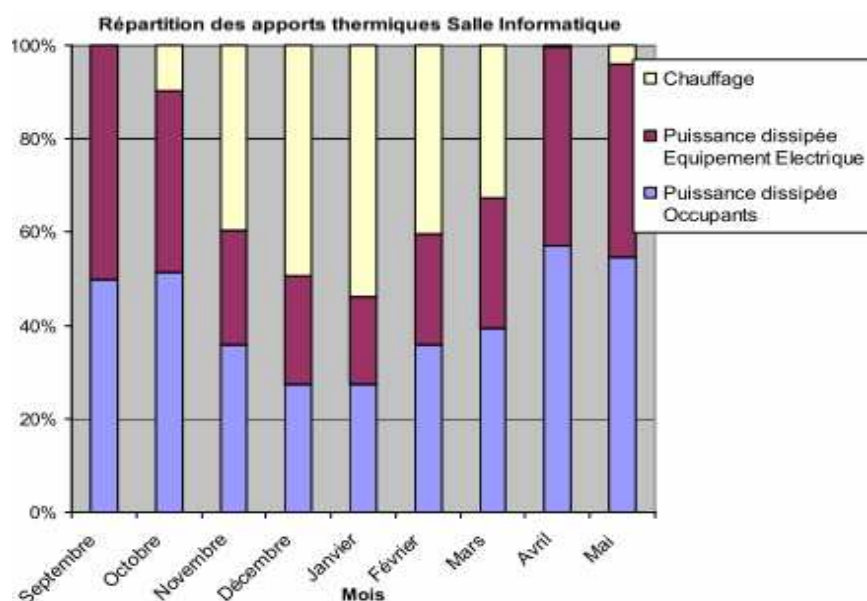


Figure 16. Quantification de la part des apports internes dans la contribution au chauffage d'une salle du bâtiment PREDIS [CHEN 2010]

Il devient donc indispensable de travailler à quantifier et minimiser ces apports internes dès la phase de conception architecturale [DUPR 2006] et de les gérer au mieux durant l'exploitation du bâtiment.

D'où l'intérêt d'une simulation dynamique globale permettant de quantifier les impacts de ces apports sur l'évolution des températures du bâtiment pour évaluer son confort d'été en plus de ses besoins de chauffage.

II.2.3. Les bâtiments intelligents (BI):

Nous avons noté précédemment que le bâtiment est un nœud énergétique complexe, dont le fonctionnement fait intervenir plusieurs acteurs, qui nécessite d'être mieux maîtrisé et dont le fonctionnement dépend de plusieurs composantes intérieures (usagers, équipements...) et extérieures (soleil, marché de l'énergie...), qui doivent eux-mêmes être contrôlés en permanence. Ces fonctionnalités dépassent les capacités humaines au niveau des moyens à mettre en place pour collecter, stocker et comparer les informations et prendre les

décisions en fonctions de plusieurs paramètres à la fois...Elles seront donc assurées par des automates et des systèmes intelligents de contrôle et de gestion de l'énergie pouvant être installés au niveau du bâtiment le rendant ainsi intelligent « Energy Smart Home ».

Il s'agit des Systèmes de Gestion Energétique du Bâtiment (SGEB) qui ont vu leurs apparitions aux Etats-Unis dans les années 1970 [NIKO 2004]. Ensuite le concept de bâtiment intelligent s'est consolidé dans les années 1980 avec les développements des nouvelles technologies et de l'informatique. Ha [HA 2007] proposait une définition précise des systèmes énergétiques comme «un ensemble d'équipements dotés de microcontrôleurs ayant des capacités de communication via des protocoles standards, un système de contrôle commande centralisé et une interface homme machine permettant de réaliser certaines fonctions d'optimisation, de conduite et de suivi de la consommation d'énergie».

Un système de gestion énergétique est donc capable de :

- Percevoir l'information grâce à des capteurs de mesure de plusieurs variables du bâtiment et des solutions d'acquisitions de données.
- Raisonner à l'aide d'algorithmes de contrôle commande et d'optimisation.
- Agir sur son environnement via les différents systèmes d'automatisation et actionneurs installés au niveau du bâtiment global ou au niveau de chaque pièce.

Un SGEB est installé dans un bâtiment dans le but de fournir plusieurs services à ses occupants comme le confort, la gestion de l'énergie, la sécurité... [ABR 2008]. Cette automatisation permet d'éviter les erreurs et les oublis humains en assurant deux fonctions principales :

- **Maîtrise de l'échange énergétique avec le réseau**: Le SGEB, devrait aider son « smart building » à mieux maîtriser son interaction avec le réseau en extraction et en injection de l'énergie. Il doit permettre d'appeler l'électricité par le chemin le plus court participant ainsi à la fiabilisation du réseau électrique et à la réduction des pertes liées au transport et à la distribution (10%). Il permet également de maîtriser la production locale d'énergie renouvelable en favorisant l'autoconsommation pendant les heures de pointe tout en tenant compte des conditions météorologiques pour leurs productions. Il détermine également les périodes optimales de vente de l'énergie produite en fonction des fluctuations du prix de l'énergie. Il permet ainsi de réaliser un retour sur investissement rapide et participe à l'équilibre de production/consommation en fournissant l'énergie au réseau lorsqu'il en a besoin [WAR 2011].

- **Pilotage des charges** : cette technique, qui va permettre de moins consommer en période de pointe, consiste à reporter les charges commandables (essentiellement électriques) en les décalant à une période ultérieure où le tarif de l'énergie est moins cher pour éviter les tarifs horaires élevés [BOED 2009]. Grâce à cette technique, le déclenchement du ballon d'eau chaude, par exemple, est programmé vers 2h du matin (heures creuses). Elle consiste aussi au délestage de certains appareils non prioritaires tel que la machine à laver pour éviter le dépassement de l'abonnement électrique et ainsi la coupure de courant [LE 2008].

Un système de gestion de l'énergie est un système complexe rassemblant plusieurs fonctions comme on peut illustrer à travers l'outil de gestion de l'énergie « G-HomeTech »

[HA 2007] [MISS 2012] issu des travaux de Stéphane PLOIX, chercheur au laboratoire GSCOP². Il est composé de trois couches (Annexe 1):

- anticipative permettant la prédiction des acteurs internes (usages, équipements) et externes (coûts, météo),
- réactive pour la correction et l'ajustement afin de respecter les anticipations vis-à-vis d'aléas,
- une couche locale au niveau des équipements pour la prise des consignes.

Par conséquent, les SGEB ont besoin non seulement d'équipements intelligents et d'algorithmes avancés d'optimisation, mais aussi de modèles de tous les acteurs pour pouvoir prédire le fonctionnement du système.

Les systèmes de Gestion Technique Centralisée (GTC) ou Gestion Technique du Bâtiment (GTB), installés principalement dans des bâtiments tertiaires font partie de ces SGEB. Ils permettent principalement la gestion des équipements de confort tel que le chauffage, la ventilation et l'éclairage...Un système de GTC a été installé dans le bâtiment PREDIS (cf chapitre 2) dans le but de superviser et contrôler finement le système de Ventilation Mécanique Contrôlée (VMC).

II.2.4. Conclusion

Ayant conscience de l'enjeu énergétique majeur du bâtiment et poussé par le désir de réduire les factures énergétiques et celui d'améliorer le confort des occupants, plusieurs mesures et orientations ont été prises pour améliorer l'efficacité énergétique du bâti à travers le label BBC et l'introduire de l'intelligence par l'utilisation des systèmes de gestion énergétique.

Ce changement de contexte énergétique et ces progrès technologiques ont des influences directes sur la modélisation et la simulation du bâtiment. Des nouvelles composantes et variantes doivent être introduites pour répondre aux nouvelles exigences.

² GSCOP: Laboratoire des Sciences pour la Conception, l'Optimisation et la Production de Grenoble : <http://www.g-scop.grenoble-inp.fr/>

II.3.Des nouveaux besoins en simulation du bâtiment

II.3.1. Les intérêts de la simulation du bâtiment

Comme tout système physique, le cycle de vie du système bâtiment se compose en quatre phases: la phase de conception, la phase de mise en place (fabrication, construction, installation, rénovation...), la phase d'exploitation et d'utilisation pour arriver à la phase finale de fin de vie (destruction, recyclage,...).

La simulation, capable de donner un comportement approché des systèmes complexes à partir de scénarios et configurations bien définis, peut intervenir dans plusieurs phases de cycle de vie dont principalement la phase de conception et la phase d'exploitation.

- Conception : elle représente un outil d'aide à la décision permettant de choisir, parmi plusieurs, une solution optimale de conception répondant au cahier des charges (la réglementation thermique en particulier). Elle intervient à plusieurs niveaux de conception :
 - o architecturale : il s'agit de choisir l'orientation du bâtiment, de la position et des dimensions des fenêtres et des portes, choix des matériaux utilisés,... de manière à satisfaire au critère de la RT relatif à l'efficacité énergétique du bâtiment (Bbio).
 - o des systèmes énergétiques : sélection et dimensionnement des systèmes énergétiques performants pour répondre aux exigences de la consommation énergétique du bâtiment (Cep) de la RT.
 - o des systèmes et algorithmes de contrôle commande : le design des algorithmes de contrôle commande pour permettre une gestion efficace des flux énergétique dans le bâtiment et une utilisation optimale de ces équipements permettant de garantir le confort des occupants dont le confort d'été (TIC).
- Exploitation : à cette étape, la simulation peut être utilisée pour :
 - o la prédiction en « temps réel » du comportement du système vis-à-vis de sa commande pour mettre en œuvre des stratégies de pilotage optimal anticipatif et réactif [HA 2007] [MISS 2012].
 - o la vérification, la validation et l'amélioration des choix de la phase de conception: il s'agit de valider les choix architecturaux, vérifier l'adaptation et le bon fonctionnement des systèmes installés ainsi que des algorithmes de contrôle commande en vue de les améliorer et de mieux les adapter au bâtiment réel.

Dans l'étude du bâtiment PREDIS que nous menons dans cette thèse, nous nous positionnons dans ce dernier cas. Nous avons développé divers modèles, afin d'aboutir à la simulation du système global, dont certains étaient initialement dédiés à la phase de conception comme le modèle de l'enveloppe COMFIE. Mais, le bâtiment PREDIS étant déjà conçu par un bureau d'étude et livré avec ses systèmes énergétiques et son système de gestion, on se propose plutôt ici d'exploiter les modèles pour, par exemple :

- améliorer les algorithmes de contrôle commande et ainsi maîtriser les consommations énergétiques.
- tester de nouveaux équipements tels que l'installation « virtuelle » d'une pompe à chaleur (thèse Hoang Anh DANG).
- étudier la sensibilité (thèse Mathilde GRANDJACQUES) de notre systèmes vis-à-vis de ses diverses sollicitations externes (météo) et internes (occupants, équipements).
- étudier un placement optimal de capteurs permettant d'identifier au mieux un modèle réduit (thèse Audrey LE MOUNIER) à des fins de pilotage optimal.
- etc.

La simulation virtuelle s'avère ainsi une étape indispensable dans la mesure où l'étude expérimentale d'un système de bâtiment se trouve confronté à [FRIT 2004] :

- Une expérimentation coûteuse en terme de temps, de mise en place et d'argent. La comparaison des performances des matériaux et des techniques de construction par exemple nécessite la construction de plusieurs projets ce qui entraîne des dépenses importantes.
- Des constantes de temps physiques très grandes nécessitant parfois des années d'expérience pour pouvoir observer des changements et obtenir des résultats, comme l'inertie du bâtiment, les conditions climatiques estivales et hivernales.
- L'inaccessibilité des variables du système réel par les mesures (e.g. champ de températures, flux thermique au travers une paroi, champ de vitesse d'air, température ressentie, ...).
- L'impossibilité de modifier dynamiquement le système : le prototypage virtuel vient remplacer le prototypage réel, peut réaliste dans le domaine du bâtiment. En effet, la simulation permet la création successive de prototypes virtuels dans lesquels varient les paramètres de fabrication (dimensions, matériaux,...) de manière à aboutir à la configuration optimale.

II.3.2. Les nouveaux besoins de simulation du bâtiment

Dans le paragraphe I.2 et I.3, on a montré l'importance de la partie électrique dans le secteur du bâtiment. Ce domaine ne doit donc plus être négligé dans la simulation et doit être modélisé de manière détaillée prenant en compte tous ses phénomènes pour mieux les étudier. Une modélisation simplifiée, représentant le comportement global des équipements électriques ne permet par exemple pas de gérer un pilotage de charge.

Dans le paragraphe II.2.2, nous avons mis l'accent sur l'importance des charges internes (usagers, appareils...) dans un contexte de BBC où l'isolation thermique est devenue très performante. Elles sont devenues plus influentes sur la gestion énergétique du bâtiment et doivent être prises en compte dès la phase de conception vu qu'elles permettent de :

- Réduire la facture énergétique en participant au chauffage d'hiver
- Augmenter la facture énergétique en impactant le confort d'été

La configuration d'un système de gestion énergétique adapté à un bâtiment permettant, comme vu dans le paragraphe II.2.3, de diminuer les consommations énergétiques

et d'améliorer le confort, nécessite une modélisation du système global tenant compte de tous ses acteurs pour une simulation performante. Or, les bâtiments actuels font cohabiter plusieurs acteurs issus de plusieurs domaines très différents nécessitant la collaboration entre plusieurs spécialités. En outre, ces parties interagissent ensemble de manière plus ou moins forte (Figure 17).

Il est donc indispensable de réaliser une simulation complète du bâtiment qui doit représenter les différents acteurs en prenant en compte des particularités de chaque domaine ainsi que des interactions qui les relient.

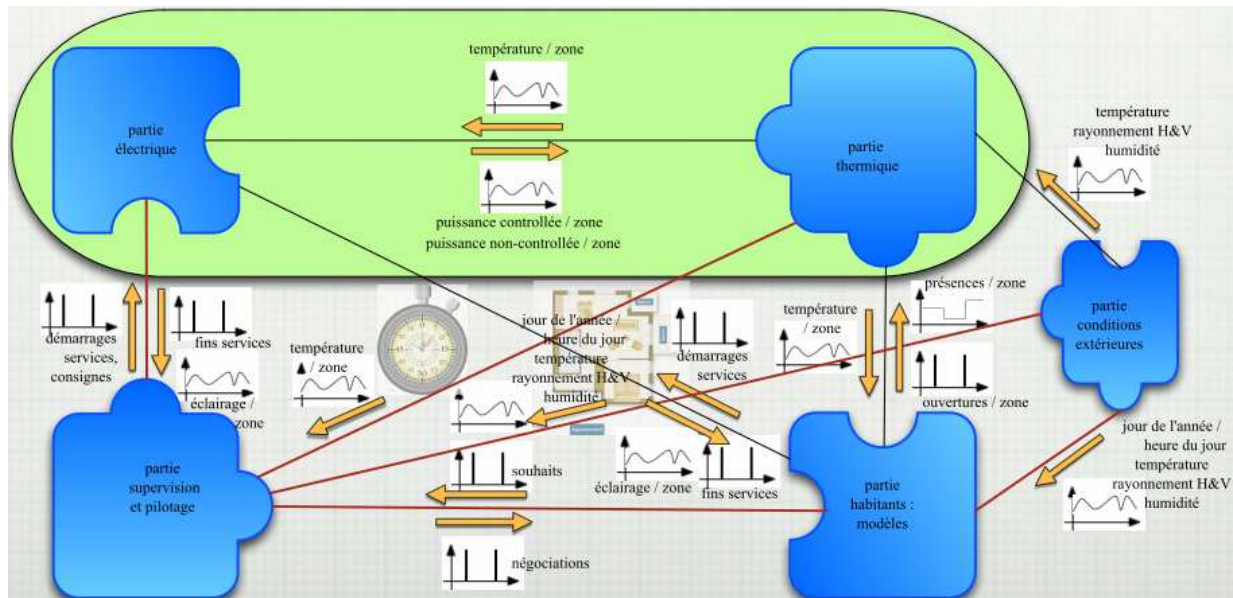


Figure 17. Les domaines interagissant dans le système bâtiment [SIMINTHEC 2009]

La réalisation de cette simulation complète du système « Bâtiment », nécessite la modélisation préalable ou l'exploitation des modèles existants de ses différents sous-systèmes issus de différents domaines d'ingénierie. La simulation globale du système PREDIS par exemple, objectif final de cette thèse, requière la prise en compte de plusieurs composants et acteurs dans la modélisation comme : l'enveloppe thermique, les équipements thermiques et électriques, les habitants...

La simulation du système s'avère ainsi une tâche délicate, confrontée à plusieurs problèmes au niveau des modèles (III.2) et/ou des outils de simulations (III.3) et qui requière des efforts considérables pour sa mise en place. Dans la suite de ce chapitre on va détailler les problématiques relatives à la modélisation et la simulation du système de bâtiment (III) et étudier les solutions existantes pour faciliter cette tâche (IV). En perspective, nous proposons une solution se basant sur un standard de composants logiciels multi facettes (V).

III. Problématiques actuelles de la simulation dynamique du bâtiment:

III.1. Introduction

Le bâtiment est un système hétérogène couplant des sous-systèmes et des phénomènes variés issus de différents domaines d'ingénierie : thermique, électrique, aéraulique, automatique, sociologique...Un utilisateur (ingénieur bureau d'étude, chercheur...) voulant réaliser une simulation complète de ce système complexe en représentant de manière assez détaillée ces acteurs, se trouve confronté à deux problématiques principales :

- Une problématique liée aux modèles : une étape de modélisation des différents composants et acteurs précède la simulation globale du système. Du fait de la difficulté de l'élaboration de tous ces modèles, dont chacun nécessite des connaissances spécifiques, on peut avoir recours à l'utilisation des certains modèles existants afin de profiter des développements réalisés (plusieurs modèles développés avec des experts de leurs domaines). On se trouve ainsi devant un large choix de modèles de différentes natures et pouvant être décrits avec plusieurs formalismes, approches et niveaux de modélisation. Il est difficile (voire impossible) d'avoir dans la bibliothèque d'un unique outil, tous les modèles dont on a besoin pour représenter les différents domaines, pour un cas d'étude donné. D'un autre côté la sélection des modèles devient délicate et doit se faire selon les objectifs en terme d'application (simulation, contrôle, dimensionnement) et selon la compatibilité avec la plateforme hôte et l'adaptabilité avec les autres modèles auxquels ils vont être couplés.
- Une problématique liées aux simulateurs : Les différents modèles nécessaires pour la simulation globale peuvent se trouver dans des outils multiples de simulation dédiés, chacun correspondant à un type de modélisation/simulation spécifique. On se trouve ainsi confronté aux mêmes types de problèmes avec une multitude de logiciels de simulation de bâtiment avec des caractéristiques et des capacités très différentes. Ces outils sont spécialisés dans leurs domaines de compétences et on est souvent amenés à exploiter les capacités de plusieurs outils à la fois, ce qui nécessite une certaine ouverture de leurs parts pour l'échange et la collaboration.

Il devient donc indispensable d'explorer des solutions permettant de dépasser ces limites pour pouvoir coupler des modèles et des simulateurs hétérogènes et de les faire interopérer afin d'aboutir à une simulation complète des différents acteurs du système.

III.2. Hétérogénéité des modèles

III.2.1. Différentes natures temporelles

Un système dynamique, par opposition à un système statique, évolue dans le temps. Par ailleurs cette variable « temps » présente des caractéristiques importantes permettant de distinguer les modèles et de les classer en deux grandes catégories [MICH 2004]:

- Les modèles continus: ce type de modèle est caractérisé par une variation continue des variables d'états sur un intervalle fini du temps (Figure 18 - gauche). La formulation mathématique associée à ce type de modèle est celle des équations différentielles [CELL 1991] dans laquelle certaines variables, appelées variables d'état, font apparaître leurs dérivées temporelles.

On distingue deux types de description temporelle continue, basées sur les équations différentielles: les équations différentielles ordinaires ODE lorsque les dérivées des variables d'état et les sorties du modèle $Y(t)$ sont décrites explicitement (Eq 1); et les équations différentielles algébrique DAE souvent décrites sous une forme implicite et introduisant dans la résolution des contraintes algébriques liées aux sorties $Y(t)$ (Eq 2).

$$\begin{aligned} \dot{X}(t) &= f(X(t), U(t), p, t) \\ Y(t) &= g(X(t), U(t), p, t) \end{aligned} \quad \text{Eq 1 : représentation ODE}$$

$$\begin{aligned} f(\dot{X}(t), X(t), U(t), p, t) &= 0 \\ g(Y(t), X(t), U(t), p, t) &= 0 \end{aligned} \quad \text{Eq 2: représentation DAE}$$

- Les modèles discrets: ce type de modèle est régi par des variables qui ne sont définies qu'à des instants précis (Figure 18 - droite). Il peut s'agir par exemple de variables continues discrétisées, tel est le cas des systèmes échantillonnés. Mais ce changement d'état peut également avoir lieu à des instants non prédéfinis : il s'agit alors des modèles à événements discrets [ZEIG 2000].

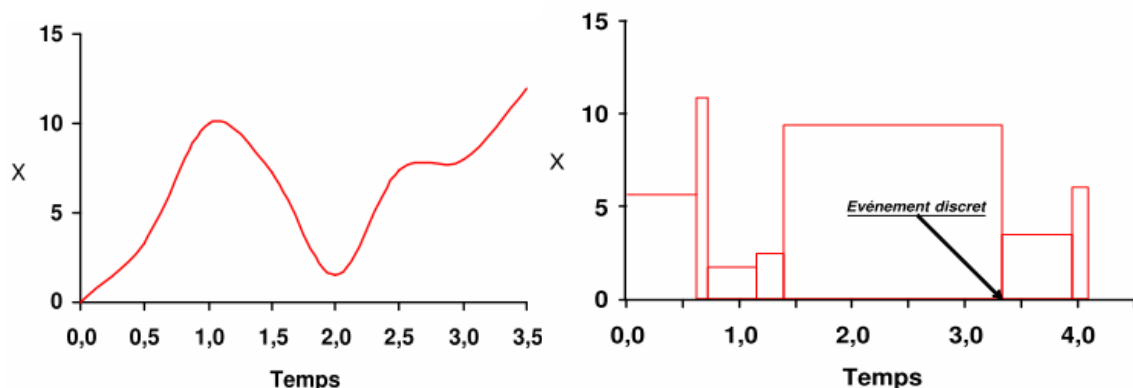


Figure 18. Représentation temporelle continue (gauche) et discrète (droite)

La combinaison de ces deux aspects nous mène à la définition des « systèmes hybrides » [HYS]. L'interaction d'un processus physique continue avec son contrôleur discret constitue par exemple un système hybride.

III.2.2. Différentes approches de modélisation

Nous allons ici définir un certain nombre de termes qui nous permettent de classifier les modèles que nous manipulons :

- Approche Analytique/ empirique (Figure 19): un modèle est dit analytique s'il est basé sur des lois et des équations physiques. Il est dit empirique s'il est issu des règles de comportement du système mesurées ou de la réduction d'un modèle analytique, la physique du modèle est donc masquée. Un modèle analytique intègre par nature plus de connaissance, et son obtention est souvent longue et complexe. Il n'est par contre pas nécessairement plus précis qu'un modèle empirique identifié pour un cas donné. En revanche, il s'adapte mieux à de nouveaux contextes d'utilisation et donc adapté à la capitalisation/réutilisation de connaissances.

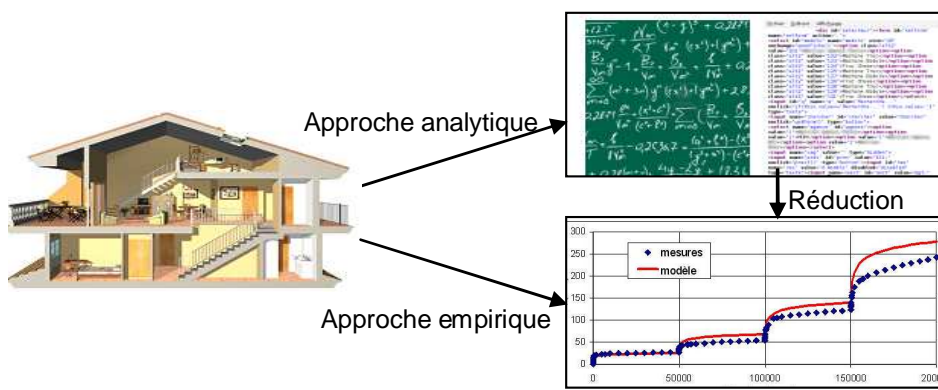


Figure 19. Approche analytique / approche empirique

- Approche « boîte noire »/« boîte blanche » (Figure 20): un modèle boîte blanche est un modèle dont la connaissance est totalement accessible et modifiable [ALL 2003]. Un modèle boîte noire, par opposition, permet d'appréhender un objet sans connaissance nécessaire sur sa constitution interne; seulement ses entrées/sorties sont accessibles. La boîte blanche permet un accès facile aux équations du modèle pour son appréhension et modification, ce qui peut être gênant en terme de confidentialité et de sûreté du modèle qui peut donc être mieux assurées par un modèle boîte noire.

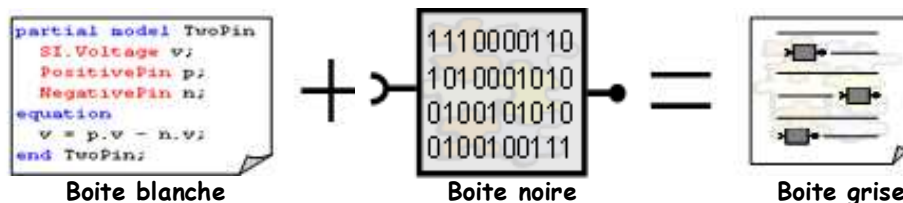


Figure 20. Approche boîte blanche / boîte noire / boîte grise

La « boîte grise » est un concept intermédiaire consistant en une combinaison des deux autres concepts dans le cadre d'un système constitué par composition.

- Approche causale/ acausale (Figure 21): l'approche causale ou « orientée » suit le principe de cause à effet entre l'entrée et la sortie : les affectations définissant une entrée doivent se situer avant celles qui la consomment [FUR 2007]. Cette

approche peut être modélisée par une boîte avec des signaux en entrée et sortie (Figure 21 - gauche).

Contrairement au modèle causal, le modèle acausal ne définit pas d'entrées et de sorties, c'est le principe de « boule » (Figure 21 - droite) [ALL 2003]. Selon Jardin [JARD 2008] « ce type de modèle consiste en un ensemble d'équations implicites non ordonnées où les entrées et les sorties du modèle (donc l'expérimentation) ne sont pas précisées » a priori et avant usage du modèle dans un processus de calcul numérique.

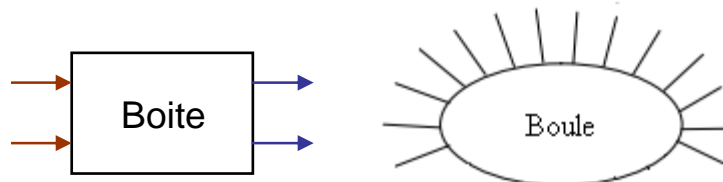


Figure 21. Approche orientée (gauche) / approche non orientée (droite)

Dans toutes ces définitions, les deux approches (boîtes/boules blanches vs boîtes noires) présentent des avantages et des inconvénients l'une par rapport à l'autre qu'on présentera en détails au fur et à mesure de leur utilisation dans les chapitres 2 et 3.

III.2.3. Différents niveaux de finesse

On peut également différencier plusieurs niveaux de finesse de modélisation. Plus le modèle est fin, mieux il représente la réalité mais plus il est compliqué à réaliser et long à résoudre par la suite, en particulier quand la taille du système est importante. Un compromis reste donc à faire selon les objectifs de l'application finale et le niveau de précision et la rapidité de calcul exigés. On peut distinguer deux types de compromis à gérer :

- un compromis finesse/coût de calcul : plus le modèle est fin moins il est rapide.
- un compromis finesse/précision de calcul (Figure 22, courbe - haut): l'augmentation de la finesse du modèle n'est pas une garantie de l'augmentation monotone de sa précision [HEN 2012]. Plus le modèle est fin, plus le nombre de paramètres croît. Ces paramètres sont associés à une incertitude qui, mises bout à bout, peuvent induire des erreurs supplémentaires. Il peut, donc, y avoir un optimum.

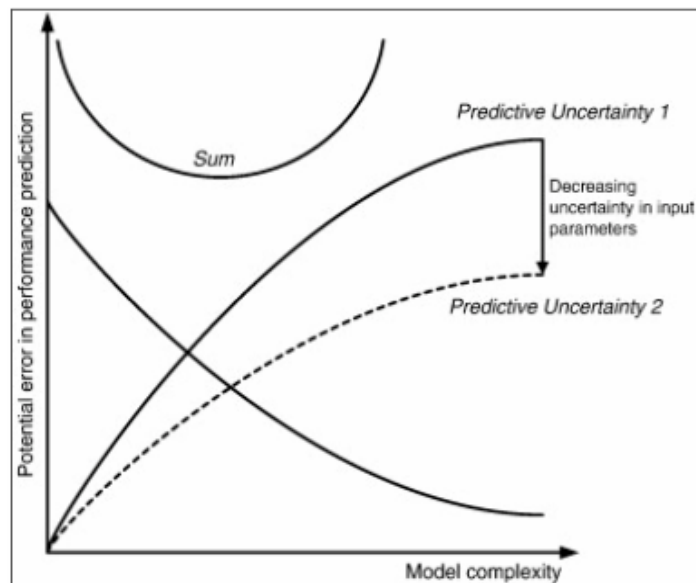


Figure 22 L'erreur dans la performance de prédiction vis-à-vis de la complexité du modèle et la notion de compromis [HEN 2012] [TRC 2010]

Le degré de granularité permet de classer les modèles selon un ordre décroissant [ELH 2006]: macroscopique, mésoscopique et microscopique.

- Niveau macroscopique : On s'intéresse ici au comportement global du système étudié ce qui ne requiert pas un modèle très précis. Un modèle empirique extrait souvent de la littérature est utilisé pour des applications comme la prédiction et la régulation.

- Niveau mésoscopique : C'est un niveau intermédiaire entre le macroscopique et le microscopique. Il s'agit d'étudier les éléments du système ainsi que leurs interactions.

- Niveau microscopique : c'est une modélisation détaillée de la physique des éléments du système.

En mécatronique, on peut prendre l'exemple d'un contacteur [REZ 2012b] où plusieurs niveaux peuvent être abordés allant de la modélisation fonctionnelle du système global, à l'étude de chaque élément (résistances, bobines...) et à l'analyse par éléments finis du champ magnétique. Dans le domaine de la modélisation thermique et aéraulique du bâtiment, on retrouve le même classement reparti principalement en [MORA 2003] :

- Modélisation mono-zone ou approche nodale (niveau macroscopique): C'est une méthode simplifiée qui a été adoptée au début des années 80 [SAL 2005] pour caractériser les transferts entre l'intérieur et l'extérieur du bâtiment. Il était considéré comme une seule zone thermique : une zone thermique et un sous-système du bâtiment ayant des caractéristiques internes uniformes. Des outils, tels que celui de la réglementation thermique ou CODYBA [ROUX 1984], implémentent cette approche.

- Modélisation multi-zone ou approche zonale (niveau mésoscopique) : L'introduction des systèmes de HVAC, par exemple, spécifique à chaque pièce, nécessite la division du bâtiment en plusieurs zones. Cette présentation considère le bâtiment comme un ensemble de zones idéalisées chacune autour d'un nœud principal. Elle permet d'étudier, en plus par rapport à l'approche mono-zonale, les transferts entre les différentes zones. Des outils phares dans la simulation du bâtiment, tel que TRNSys, utilisent ce type de modélisation.

- Modélisation par éléments finis (niveau microscopique): Il s'agit de décomposer les volumes des zones en mailles très fines. On peut ici citer les techniques de calcul de champ : CFD (*Computational fluid dynamics*) [KUZ 2010] implémentés dans le logiciel Fluent.

Le rapport du projet Dynasimul [DYNA 2010] met l'accent sur les difficultés de couplages entre des modèles avec des niveaux de finesses différents à cause des disparités sur l'échelle de temps, d'espace et de vitesse de calcul. Il présente également des techniques et des cas test de couplages.

III.2.4. Différents dimensions spatiales

En plus de la dimension temporelle qui permet de différencier les modèles statiques (0D) et dynamiques (1D), on retrouve également la dimension spatiale. On peut ainsi distinguer plusieurs typologies de modèles : 0D, 1D, 2D et 3D (Figure 23).

- Des modèles 0D : aucune paramètre spatiale n'est prise en compte.
- Des modèles 1D : une seule dimension spatiale est traitée dans les équations aux dérivées partielles. Tel est le cas d'un modèle d'enveloppe de bâtiment dans Comfie. En effet, l'enveloppe est un assemblage de parois, de fenêtres, de portes dans lesquels le flux thermique est supposé unidirectionnel.
- Des modèles 2D/3D : Les modèles précis prennent en compte des effets 2D voire 3D (plus rare en thermique). Le cas des ponts thermiques doit par exemple être traité dans plusieurs dimensions d'espace.

Pour conclure, les modèles peuvent donc être placés sur le diagramme suivant selon leur prise en compte de phénomènes temporels et spatiaux.

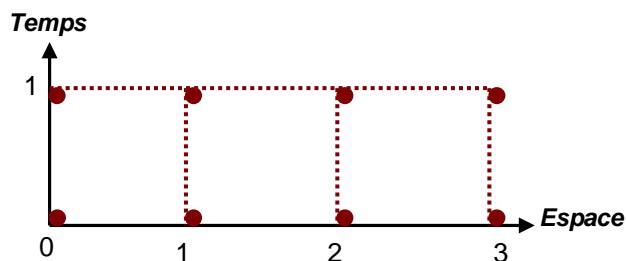


Figure 23 Les dimensions spatio-temporelles

III.2.5. Les difficultés liées à cette hétérogénéité et les besoins d'interopérabilité

En étudiant les caractéristiques principales des modèles utilisés pour la simulation des bâtiments, plusieurs disparités sont observées, au niveau de leurs natures temporelles, de leurs approches et formalismes de description, ainsi que leurs niveaux de finesse de modélisation.

Cette hétérogénéité est due généralement, selon Hardebolle [HARD 2008], à une différence dans le domaine d'application pour lequel le modèle est destiné (électrique, thermique...) et dans les objectifs pour lesquels il a été conçu (dimensionnement, contrôle, simulation...). Un modèle peut dépendre également de l'environnement dans lequel il a été développé lui imposant un certain nombre de contraintes.

Cette hétérogénéité s'avère ainsi problématique si on veut :

- Utiliser un modèle pour d'autres fins et objectifs, nécessitant un effort d'adaptation pour correspondre aux nouvelles utilisations.
- Utiliser le modèle dans d'autres environnements, formalismes et technologies de description, nécessitant la réalisation de multiples transformations et reformulations qui peuvent s'avérer lourdes (temps, effort) et présente des risques de perte de robustesse du modèle (faire des tests de validation).
- Composer le modèle avec d'autres modèles différents (nature, domaine...) constituant le système global. Cette composition nécessitera l'étude de leur compatibilité et exigera parfois d'apporter des modifications aux modèles initiaux (réductions, approximations, redéfinitions des entrées et des sorties des modèles...).

Nous avons très concrètement rencontré cette hétérogénéité lors de la modélisation du système PREDIS (cf chapitre 2). Des modèles statiques (échangeurs de chaleurs) et dynamiques (enveloppe), des modèles continus, discrets (enveloppe) et événementiels (occupants, chauffage), des modèles analytiques et empiriques, des modèles multi zones et différences finis...ont été élaborés, récupérés et adaptés. On se trouve amené à utiliser des modèles pour des fins différentes de celles pour lesquelles ils étaient initialement définis, les projeter dans d'autres environnements et formalismes de description et les coupler à plusieurs autres modèles pour pouvoir aboutir à une modélisation globale du système.

***Pour remédier à cette hétérogénéité,
il est indispensable d'adopter des solutions d'interopérabilité
garantissant une adaptation et une transformation aisées et automatiques des modèles
ainsi qu'une plus large compatibilité avec d'autres modèles et outils de simulation.***

III.3. Problématiques liées aux simulateurs

III.3.1. La multiplicité des outils de simulations

La simulation du bâtiment devient de plus en plus variée et hétérogène impliquant plusieurs composants de différents domaines d'ingénierie. Des centaines d'outils de simulations ont été développés pour aider les architectes, les concepteurs de bureau d'études et les chercheurs à travers le monde entier et les suivre dans les différentes phases du projet d'un bâtiment.

Le département de l'énergie des Etats Unis [DEEU 2011] a recensé en 2011, 404 outils de simulation reliés à l'évaluation de l'efficacité énergétique, les énergies renouvelables et le développement durable... Il s'agit d'outils de simulation des performances énergétiques du bâtiment (BPS : Building Performance Simulation).

Ce chiffre peut facilement augmenter si on rajoute à la liste, les logiciels de simulation spécialisés dans d'autres domaines d'ingénierie et pouvant être utilisés pour une modélisation avancée de certains équipements du bâtiment.

Le nombre des outils de BPS plus particulièrement augmenté ces quinze dernières années. Il a été multiplié par quatre en passant d'une centaine d'outils en 1997 à 389 en 2010 (Figure 24).

Cette remarquable évolution est due à une conscience des enjeux majeurs de ce secteur qui implique une maturité dans le domaine de la simulation du bâtiment [ATT 2010]. En effet, c'est à partir des années 90 que commence le changement de mentalité, où nous sommes passé de la seule préoccupation de la performance thermique et de la consommation énergétique (calcul de charge et analyse de la consommation) à la prise en compte de plusieurs autres performances [AUG 2002] tel que le confort de l'occupant, le transfert des flux d'air, l'aspect acoustique...

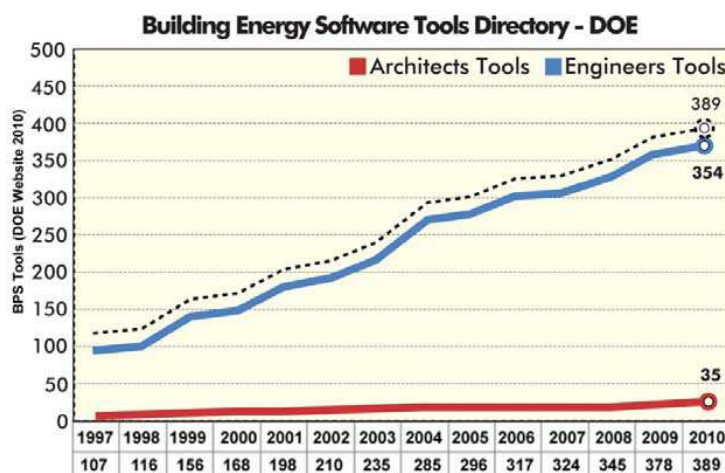


Figure 24. Les outils de BPS développés entre 1997 et 2010 [ATT 2011]

Une multiplication des outils implique une couverture plus large de la diversité des modèles nécessaires, mais n'induit pas nécessairement une plus grande aisance de modélisation. Au contraire, cette variété peut représenter une barrière au moment de la

sélection des outils face à ce large choix. Des problèmes d'indécision lors de la sélection peuvent être évoqués étant donné :

- Les grandes similitudes entre plusieurs outils.
- Le fait que les outils mentionnent rarement les limitations de leurs capacités. Dans ce cas, seule, l'expérience permettra de les déterminer.
- La possibilité de variation de la sélection des outils selon les critères choisis. Donc les benchmarks réalisés pour aider l'utilisateur à la décision ne correspondent parfois pas aux critères des utilisateurs.

Parmi les benchmarks réalisés pour faciliter les choix des outils de simulation, on peut citer le rapport du département de l'énergie publié en 2005 [CRAW 2005], qui compare les vingt outils les plus utilisés internationalement.

Au niveau de la France, le rapport Simbio [MORA 2009] comparait les outils de simulation dynamique des performances énergétiques les plus significatifs pour le marché français en partant des critères semblables au précédent rapport: ENERGYPLUS³, ESP-R⁴, TRNSYS [WEB_TRN], SIMBAD [WEB_SIM], COMFIE+PLEIADES [WEB_COM], TH-C-E/TH-C-E-X⁵, CODYBA/CODYRUN⁶ et SIMSPARK⁷.

III.3.2. La non adéquation d'outils de simulation à de nouveaux besoins

Le bâtiment est un secteur en permanente évolution ce qui peut impliquer que des outils largement utilisés à une époque donnée, peuvent être rejetés plus tard à cause de leur incapacité à suivre les évolutions du secteur.

Plusieurs outils peuvent ne plus être adaptés à la simulation énergétique des bâtiments pour diverses raisons:

- Ils ne permettent pas de tenir compte des nouveaux contextes énergétiques (bâtiments basses consommation, bâtiments à énergie positive...) et les nouvelles contraintes environnementales qui y sont reliées. Ils ne permettent pas par exemple la vérification du respect des critères de la nouvelle réglementation thermique et de créer le diagnostic de performance énergétique (DPE).
- Ils ne prennent pas en compte les nouvelles évolutions du domaine nécessitant par exemple l'implication des différents utilisateurs aux différentes phases du projet (dès la phase de design). Cela peut concerner jusqu'aux architectes qui exigeront de plus en plus d'outils de simulation performants, mais ne nécessitant pas de hautes compétences techniques.
- Ils n'intègrent pas les dernières avancées technologiques et les nouveaux systèmes énergétiques qui sont de plus en plus présents dans les bâtiments tels que les systèmes de VMC doubles flux, les matériaux à changement de phase, les sources d'énergie renouvelables (photovoltaïque, éolienne...)...

³ <http://apps1.eere.energy.gov/buildings/energyplus/>

⁴ <http://www.esru.strath.ac.uk/Programs/ESP-r.htm>

⁵ <http://www2.equipement.gouv.fr/bulletinofficiel/fiches/Bo200818/A0180009.htm>

⁶ <http://www.jnlog.com/codyba1.htm>

⁷ http://simspark.sourceforge.net/wiki/index.php/Main_Page

- Ils ne répondent pas aux nouveaux besoins de simulation globale du bâtiment qui requièrent la prise en compte de tous les domaines concernés. On se trouve par exemple devant des outils des années 60, 70 se focalisant essentiellement sur les performances thermiques, le calcul de charge... [KUS 1999].

Les exigences des utilisateurs et de l'environnement sociotechnique ne cessent de croître de manière à toujours dépasser les capacités des outils de simulation actuels [HEN 2000] poussant toujours à l'innovation et au développement de nouveaux outils. De leurs côtés, les outils doivent suivre ces évolutions dynamiquement et répondre aux exigences en incessante évolution.

III.3.3. La spécialisation des outils de simulations

III.3.3.1. Limitations des outils existants

Des nombreux outils de simulations ont été développés dans le secteur du bâtiment. Cependant il n'existe aucun outil qui permet à lui seul de réaliser:

- la simulation détaillée de tous les domaines du bâtiment : thermique, électrique, contrôle commande, comportement d'utilisateur...
- l'analyse et la résolution des différents types de problèmes rencontrés : matériels, logiciels, détection de défaillances...
- Intervenir à toutes les phases du projet traité: design, exploitation, vérification, recyclage...

Ainsi, un outil peut couvrir plusieurs métiers, applications et domaines d'utilisation mais il ne peut jamais être complet et répondre inconditionnellement à toutes les exigences et simuler les différents composants dans toutes les phases de vie du bâtiment.

Chaque outil est généralement dédié à des applications spécifiques : il peut servir à la simulation du bâtiment dans son ensemble tel que COMFIE ou à l'analyse des équipements et des systèmes tel que TRNSYS (le système de HVAC par exemple), comme il peut être dédié à des tâches plus précises de calcul économique et évaluation de la pollution...

Une étude préalable des outils de simulation devient donc nécessaire pour sélectionner les simulateurs qui répondent aux besoins d'utilisation.

III.3.3.2. Illustrations au travers des outils étudiés

Nous revenons à notre application principale visant à réaliser une simulation globale du bâtiment PREDIS. Pour satisfaire à nos besoins en terme de modélisation et de simulation complète de ce système tenant compte de ses différents acteurs (enveloppe thermique, systèmes énergétiques, système de supervision, occupants...), nous avons sélectionné plusieurs outils qui permettent de couvrir plusieurs domaines. Nous allons aborder et utiliser essentiellement COMFIE qui est dédié à la modélisation de l'enveloppe, TRNSYS qui permet la simulation des systèmes énergétiques du bâtiment, Matlab / Simulink pour le contrôle commande et le test des stratégies avancées de simulation (ou co-simulation), Modelica / Dymola pour la description des systèmes énergétiques (VMC double flux) et Brahms pour la simulation du comportement des occupants.

On présente ainsi les caractéristiques de ces outils qui ont été utilisés durant mes travaux de thèse en insistant sur leurs points forts pour justifier leurs choix de sélection et

certaines de leurs limites incitant à employer d'autres solutions complémentaires (d'autres outils et des solutions d'interopérabilités):

COMFIE [PEUP 1990]:

Ce logiciel de simulation dynamique est dédié aux ingénieurs et aux architectes afin de leur fournir des estimations précises des besoins énergétiques et des profils de température.

Les données saisies sont: la géométrie du bâtiment, les caractéristiques thermiques, les charges internes et leurs scénarios d'usage, le climat ainsi que les caractéristiques des équipements de chauffage et de ventilation. A partir de ces données, Comfie calcule les charges annuelles et horaires dus au chauffage ainsi que les températures horaires et moyennes des zones thermiques.

Il offre aussi une modélisation détaillée de l'enveloppe, contrairement au calcul mono zonal de la RT, se basant sur une méthode aux différences finies en y appliquant une technique de réduction modale pour réduire les temps de simulation. Ceci a permis à COMFIE d'être à la fois précis et rapide avec quelques secondes ou quelques minutes de simulation pour un projet complexe. Ceci lui a permis une large utilisation chez les architectes et les bureaux d'étude en France. Cette simulation dynamique précise est indispensable dans l'évaluation de l'influence de l'inertie sur les deux variantes de confort d'été et de consommation hivernales au niveau des bâtiments basse consommation.

Par contre, les équipements et les usagers y sont modélisés de manière simpliste : des scénarios prédéfinis de puissance maximale, consigne et position du thermostat pour le chauffage. Il est aussi impossible d'introduire des nouveaux modèles ou d'améliorer les modèles existants.

DYMOLA (Modelica) :

DYMOLA [WEB_DYM], DYnamic MOdelling LABoratory, est un environnement de modélisation et simulation dynamique se basant sur le langage Modelica.

Modelica [WEB_MOD] est un langage multi physique de modélisation comme plusieurs autres langages VHDL-AMS [WEB_VHD], Verilog-AMS⁸, Simscape⁹... Modelica est un langage orienté objet de modélisation des systèmes physiques dynamiques et multi-domaines tel que des systèmes de la mécatronique, l'automobile, l'aéronautique....

C'est un langage émergent dans le secteur du bâtiment et plusieurs travaux l'ont mis en avant ces dernières années conduisant au développement de multiples bibliothèques de composants. En plus de la bibliothèque standard [WEB_LIB] qui couvre plusieurs domaines d'ingénierie, on peut citer la bibliothèque *ATplus* [FEL 2002] pour la simulation du comportement thermique du bâtiment, *HumanComfort* [MIC 2009] des modèles de prédiction du confort de l'occupant ou également la bibliothèque Buildings [WET 2009] pour les systèmes énergétiques et de contrôle dans le bâtiment.

Il s'agit d'un langage libre qui a été développé par l'association de Modelica lancée depuis 1996 et pouvant être simulé dans des environnements dédiés :

⁸ <http://www.eda.org/verilog-ams/>

⁹ <http://www.mathworks.fr/products/simscape/>

- Libres : tel que OpenModelica¹⁰, JModelica¹¹, Modelicac de SCICOS¹².
- Commerciaux : tel que Dymola, LMS AMESim¹³ ...

L'environnement Dymola a été sélectionné pour plusieurs raisons :

- Une large utilisation dans le secteur du bâtiment
- Une interface graphique très conviviale
- L'implémentation d'une grande partie des spécifications de Modelica, des nouvelles fonctionnalités telle que les FMI [WEB_FMI] [DYMO 2010] avec ses différentes versions.
- Utilisation des traducteurs puissants capables de réaliser les transformations symboliques de grands systèmes (> 100 000 équations) et des solveurs performants.

Matlab / Simulink (Mathworks) :

Matlab [WEB_MAT] est un environnement de développement dédié à des calculs numériques puissants couvrants des applications variées dont essentiellement la conception des systèmes de commande, le traitement du signal et d'images...

Simulink est une plateforme, intégré dans Matlab, de modélisation et de simulation multi-domaine pour les systèmes dynamiques et embarqués. Il fournit un ensemble de bibliothèques avec plusieurs blocks de modèles permettant la conception et la simulation de systèmes dynamiques. Il dispose de block S-function permettant l'introduction des nouveaux modèles dynamiques et offre des interfaces de communication directe avec le solveur.

BRAHMS (Agent iSolutions) :

Brahms [WEB_BRA] est une plateforme pour la simulation multi agents du comportement des personnes et des machines. Il a été initialement développé pour des objectifs d'analyse des organisations des personnes et des processus de travail. Le comportement humain est basé sur une structure BDI (Belief, Desire, Intention), en d'autres termes un principe cause - effet. Un agent (occupant par exemple) dans Brahms est défini, dans le cadre d'un groupe d'agent, par ses activités (exemple : déplacement) qui peuvent être en lien avec d'autres objets, son emplacement géographique...

Il dispose d'un composeur pour la description des modèles, d'un compilateur pour les analyser, d'une machine virtuelle (VM) pour exécuter les modèles et d'un outil de visualisation qui permet de tracer l'historique des événements et de visualiser dans un axe temporel les activités des agents, leurs mouvements et interactions.

III.3.4. Besoins d'interopérabilité entre les outils de simulation du bâtiment

On peut résumer les problématiques liées aux simulateurs du bâtiment ainsi :

- Un secteur en permanente évolution imposant de pouvoir répondre dynamiquement aux exigences et aux besoins qui en dérivent.
- Des capacités variées et en même temps limitées à des spécialités et des phases de projet données.

¹⁰ <http://www.openmodelica.org/>

¹¹ <http://www.jmodelica.org/>

¹² <http://www.scicos.org/>

¹³ <http://www.lmsintl.com/LMS-Imagine-Lab-AMESim>

Pour dépasser les limites de spécialisation des outils de simulations existants et pour répondre aux exigences en incessante évolution des utilisateurs, il devient nécessaire d'utiliser des solutions d'interopérabilité permettant d'améliorer la modularité des outils (pour en importer et exporter des fonctionnalités) et/ou d'assurer un travail collaboratif couvrant les besoins de simulation du système de bâtiment

Pour atteindre ces objectifs, un critère devient important dans le choix des outils. Il s'agit de leur capacité d'inter-opérer. Il devient aussi important que les autres critères cités par Attia [ATT 2010] tel que : interfaces conviviales, intégration des systèmes technologiques intelligents, la précision ...Il devient donc indispensable d'étudier au préalable les capacités des outils à pouvoir échanger des modèles en import et export ainsi qu'à offrir des interfaces de communication pour leurs pilotages externes.

Or, ces outils sont caractérisés par des compétences et des domaines de spécialisations différents. Etant donné que la nature des modèles développés dans ces outils est reliée au domaine traité, aux objectifs et aux environnements pour lesquels ils sont destinés, on se trouve ainsi avec des outils supportant différents types de modèles comme récapitulés dans le tableau suivant (Tableau 1):

	Boite blanche / Boite noire	Causale / acausale	Nature temporelle	Formalisme de description	Dimension spatiale
COMFIE	Boite noire	causale	discret	Pascal	1D
Dymola	Boite blanche / Boite noire (fmi)	causale / acausale	discret / continu / événement	Modelica : Orienté objet	0D
Matlab / Simulink	Boite blanche / Boite noire (dll)	causale / acausale (SimpScape)	discret / continu / événement	Matlab / c++	0D
Brahms	Boite blanche	causale	événement / discret	Brahms : Multi-agents	0D

Tableau 1. Les natures des modèles supportés dans les outils utilisés

Ces outils utilisent également des types de solveurs différents adaptés aux modèles traités pour pouvoir les simuler efficacement. Ils offrent aussi des interfaces (pour l'import des modèles, la communication avec leurs solveurs (Tableau 2)...) avec des syntaxes très variées même si les fonctionnalités se ressemblent.

Nous essayerons ainsi d'identifier une stratégie d'interopérabilité (IV) permettant un échange aisé des modèles et une communication automatisée entre les outils tout en tenant compte de leurs particularités et disparités.

Cette stratégie d'interopérabilité doit apporter des améliorations des fonctionnalités des outils actuels avec un minimum d'effort de développement en leur permettant de:

- profiter des développements réalisés dans plusieurs autres outils de simulation dans leurs domaines de compétence.
- intégrer et simuler facilement des nouveaux équipements.

- assurer la continuité entre plusieurs phases du projet : un modèle développé pendant la phase de design tel qu'un modèle de bâtiment dans COMFIE peut être utilisé pendant la phase d'exploitation pour la simulation virtuelle pour la validation ou pour la prédiction (cf chapitre 4).

IV. Les solutions existantes d'interopérabilité

IV.1. Stratégies d'interopérabilité :

Selon l'IEEE, l'interopérabilité se définit comme « *la capacité de deux ou plusieurs systèmes ou composants d'échanger l'information et d'utiliser cette information qui a été échangée* ». Les approches d'interopérabilité utilisées dans le secteur du bâtiment peuvent être classées en trois principales catégories en se basant sur les études de Hensen [HEN 2004]:

IV.1.1. Interopérabilité des données des modèles

Cette approche se focalise sur le transfert de données entre les programmes au niveau des modèles. Cette interopérabilité peut être assurée à travers deux moyens :

- Une base de données: il s'agit d'une unique base de données qui stocke et gère le transfert des informations des modèles entre les environnements de modélisation. On peut ici citer l'exemple du projet COMBINE¹⁴ [AUG 1994]. Cette stratégie nécessite une importante base de données et exige qu'à chaque modification du modèle, chaque membre en soit notifié.
- Un format standard des données : cette méthode exige l'établissement d'un format commun des données entre différentes applications pour faciliter l'échange de données entre eux et ce qui nécessite par la suite la mise en place de traducteurs pour leurs utilisations dans chaque hôte. C'est pour cet objectif, que l'IAI¹⁵ qui est une alliance entre les entreprises du secteur du bâtiment et les éditeurs de logiciels, a initié le lancement des IFC. Les IFC [BAZ 1999] est un format de fichier orienté objet largement utilisé dans l'industrie de la construction pour échanger des données entre les logiciels les plus utilisés dans les différentes phases de vie du bâtiment.

Bien que ces stratégies permettent de faciliter l'échange des informations de modèles en les créant une seule fois, évitant ainsi les erreurs lors du re-développement, elles ne peuvent cependant intervenir qu'avant la phase de simulation et donc elles ne répondent qu'à une partie des besoins d'interopérabilité. Cette stratégie d'interopérabilité n'est pas étudiée dans la suite de ce travail.

IV.1.2. Interopérabilité des processus des modèles

Ce type d'interopérabilité est assuré au niveau des processus des modèles physiques et peut être réalisé à travers deux moyens :

- L'échange et la réutilisation des composants de modèles: cet échange peut se faire au niveau du code source ou, une perspective plus intéressante, au niveau des bibliothèques pré compilées (statique ou dynamique) tel est le cas de la stratégie adoptée pour l'échange de modèles de TRNSys vers Matlab/Simulink à travers la dll correspondant au « type » de TRNSYS [REID 2009]. Il s'agit d'une approche boîte noire favorisant la réutilisation des codes des modèles mais reste nouvelle

¹⁴ COMBINE: Computer Models for the Building Industry in Europe

¹⁵ IAI: International Alliance for Interoperability

dans le bâtiment [TRC 2008] et le manque de standardisation rend son implémentation dans les différents outils difficile.

Ce besoin de standardisation des modèles a été préalablement identifiés dans d'autres communautés ce qui a conduit à la spécification de certaines normes telles que les FMI [FMI 2010a] voyant leurs origines dans le standard AUTOSAR [WEB_AUT] de l'industrie automobile. Quelques études de l'adaptabilité de certains outils de simulation du bâtiment à la norme FMI ont été lancées ce qui prouve le réel besoin d'une telle standardisation spécifique à cette communauté.

- L'utilisation d'un langage neutre et générique: il s'agit ici de définir les différents modèles du bâtiment et de ses systèmes dans le même langage comme NMF [VUOL 1999] ou Modelica [TIL 2001] qui a émergé ces dernières années dans le secteur du bâtiment. Comme toute projection vers un format standard, cette technique nécessite le développement de plusieurs traducteurs depuis plusieurs outils pour éviter les erreurs d'une traduction manuelle. Il s'agit d'une approche boîte blanche d'interopérabilité nécessitant la connaissance totale des modèles.

IV.1.3. Coopération des données et des processus des modèles

Cette approche se focalise sur les couplages des programmes en vue d'un échange mutuel d'information d'une manière prédéfinie pendant la simulation. Cette stratégie est appelée également « co-simulation » et permet plus de flexibilité de modélisation et simulation que les autres approches, étant donné que les modèles sont indépendamment créés et séparément simulés dans leurs propres outils. Dans certains cas, elle s'avère la seule solution possible d'interopérabilité, quand il est délicat de séparer les modèles et leurs données de leur solveur. En outre, les autres solutions sont préférables lorsqu'elles sont possibles car la co-simulation offre de faibles performances et d'éventuels problèmes d'instabilité numérique. A part plusieurs couplages directs réalisés entre quelques logiciels de simulation du bâtiment tel que le couplage entre EnergyPlus et TRNSys ou entre ESP-r/EARTH [TRC 2008], un effort de généralisation et d'automatisation pour faciliter la mise en œuvre d'une co-simulation a été réalisé.

Il s'agit du BCVTB [WEB_BCV] [WETT 2011], une plateforme assurant le couplage de plusieurs outils et avec laquelle plusieurs connexions ont été développées. Il s'agit ici d'une standardisation du moyen de communication : l'échange d'information via des *sockets*¹⁶. L'orchestration est assurée par l'environnement multi domaines Ptolemy pour la modélisation et la simulation de systèmes hétérogènes. Un domaine dans Ptolemy [WEB_PTO] signifie l'ensemble des règles qui permettent d'interpréter un modèle dont : le mode de gestion du temps et le mode de synchronisation et de communication entre modèles...

D'un autre côté aucun travail de standardisation des acteurs de la co-simulation n'a été répertorié dans le domaine de la simulation énergétique du bâtiment malgré l'utilité de ce concept démontré dans d'autres domaines [FMI 2010b].

¹⁶ Socket : interface logicielle avec les services du système d'exploitation, grâce à laquelle un développeur exploitera facilement et de manière uniforme les services d'un protocole réseau tel que TCP/IP

IV.2. Capacités d'interopérabilité des outils utilisés

Avant d'identifier la solution d'interopérabilité la plus adaptée à un cas d'étude donné, il est indispensable de commencer par étudier les capacités d'interopérabilité offertes par les outils qui seront utilisés. C'est ce que nous avons tenu à faire.

Les capacités des outils utilisés décrits précédemment sont classées (Tableau 2) selon la possibilité d'exporter leurs modèles ou d'importer des nouveaux modèles externes ainsi que la possibilité de fournir un modèle avec son algorithme de résolution (modèle de co-simulation):

	Import des modèles	Export des modèles	Export des modèles de co-simulation
COMFIE	non	Oui (par fichiers)	Oui (par fichiers)
TRNSYS	Oui (nouveau composant Type)	Oui (dll du Type)	Oui (TRNDll.dll)
Dymola	Oui (fichier Modelica/fonctions externes/fmi)	Oui (fichier Modelica, fichier C du code Modelica compilé, fmi)	oui (fmi co-simulation)
Simulink	Oui (S-function)	Oui (dll Embedded Coder)	Non
Brahms	Oui (par programmation)	Non	Oui (API)

Tableau 2. Les capacités d'interopérabilités des outils utilisés

Ces outils offrent des possibilités intéressantes et variées en terme d'interopérabilité. Nous en avons exploité quelques unes, selon nos besoins, au fur et à mesure de ce travail pour montrer l'intérêt de la solution d'interopérabilité adoptée.

On peut observer par exemple que le logiciel COMFIE ne permet pas d'importer des nouveaux modèles, c'est pour cela qu'il ne sera jamais utilisé comme un hôte de la solution d'interopérabilité mais comme générateur, contrairement aux outils Simulink, TRNSYS et Dymola.

Brahms par exemple n'offre pas la possibilité l'utilisation externe de ces modèles seuls. Ils doivent toujours être accompagnés de leur machine virtuelle de résolution. Donc un modèle de Brahms ne peut être utilisé que dans une co-simulation.

IV.3. Conclusion

Après une analyse des solutions existantes et des besoins du domaine en terme d'interopérabilité entre les modèles et les outils de simulation, afin de pouvoir mettre en œuvre une simulation globale du système de bâtiment, on note un manque de standardisation au niveau des API de ces acteurs. La convergence vers une solution permettant à la fois l'échange des modèles entre les outils ainsi que la coopérations des simulateurs devient indispensable. Et la standardisation des leurs interfaces va permettre de faciliter l'automatisation et la généralisation de leur réutilisation.

L'amélioration de l'interopérabilité entre modèles et outils de simulation dans le secteur du bâtiment a donc besoin d'une nouvelle solution qui prend en considération l'hétérogénéité de ces modèles, les spécificités de chaque domaine tout en unifiant les interfaces de communications. Cette solution vise à atteindre des perspectives ambitieuses permettant de:

- Couvrir plusieurs domaines de simulation du bâtiment (thermique, électrique, contrôle commande, usager...) en s'appuyant sur l'offre existante en modélisation / simulation.
- Adresser plusieurs métiers (simulation dynamique (co-simulation), optimisation, dimensionnement...) et s'auto adapter selon la « vue » de projection : on peut imaginer les équipements futurs disposer à la fois de leur modèle de prévision ainsi que leur modèle de simulation et pouvant utiliser l'un ou l'autre selon le besoin de l'utilisateur.
- Etre facilement compatible avec une large sélection de modèles hétérogènes (boîte blanche / boîte noire...) et d'outils utilisables dans différentes phases du cycle de vie du bâtiment.
- Capitalisation des connaissances via des plateformes collaboratives d'échange tel que Dimocode [WEB_DIM] et dynasimul¹⁷ et partage des modèles via Internet en exploitant la notion des web services et du « cloud computing ».

Ces perspectives ne pourront être atteintes qu'au prix d'un changement dans le paradigme actuellement appliqué aux structures des applications de conception, de simulation et de supervision de la filière. Ceci exige d'aller au delà du paradigme d'environnements, c'est-à-dire au-delà des logiciels mis en œuvre aujourd'hui dans les bureaux d'étude pour réaliser la conception d'un bâtiment, ou mis en œuvre in situ pour réaliser la gestion technique et qui s'appuient sur des environnements informatiques non distribués et non inter opérables.

Dans la suite de cette thèse (chapitre 3 et 4), nous proposons une solution d'interopérabilité permettant de satisfaire aux besoins de ce secteur et aux exigences de ces utilisateurs. Il s'agit d'une solution s'appuyant sur un nouveau paradigme de composants logiciels [SZY 1998] et vise à établir une norme des interfaces des modèles et des simulateurs.

¹⁷ <http://dynasimul.univ-lr.fr/>

V. Conclusion

Nous avons mis en évidence à travers le paragraphe II.2 de ce chapitre que le bâtiment représente un grand consommateur d'énergie électrique, un enjeu énergétique majeur et une variable d'ajustement du réseau électrique permettant de lisser la variation des profils de consommation et réduire les risques de « black-out ». Le bâtiment représente ainsi une filière de recherche intéressante à explorer en génie électrique en lien notamment avec la décentralisation de la production énergétique et l'intégration des énergies renouvelables. Les laboratoires de génie électrique, et le G2ELAB en particulier, se sont rendu compte de ces perspectives en terme de recherche et une synergie commence à se créer autour de ce secteur.

De l'autre côté, la consommation électrique devient majoritaire dans le secteur du bâtiment et la partie électrique devient de plus en plus influente (chapitre 1 : II.3). Ce domaine doit donc être plus que jamais pris en considération et étudié de plus près. Et pour mieux maîtriser ces équipements, optimiser leurs utilisations et réduire leurs consommations, des systèmes de gestion énergétique SGEB seront de plus en plus utilisés dans les bâtiments.

Ces systèmes de gestion énergétique doivent prendre en compte tous les acteurs externes (météo, prix de l'énergie) et internes (équipements, usages...) ce qui requière une simulation complète du bâtiment tenant compte de tous ces éléments et de leurs interactions (chapitre 1 : II.4).

La mise en place d'une simulation globale du bâtiment se trouve confronté à la spécialisation des outils de simulation (chapitre 1 : III.3) et leurs limitations à certains domaines et métiers. Ceci nécessite des solutions d'interopérabilité permettant une collaboration de ces outils à travers l'échange de modèles (chapitre 4 II) ou à travers leurs co-simulation (chapitre 4 III).

Or, ce domaine est caractérisé par des modèles très hétérogènes, comme on va l'observer à travers la modélisation de PREDIS (chapitre 2) et par des interfaces de communication très variées. Une première solution d'interopérabilité sera employée, dans un premier temps, se reposant sur la projection des tous les modèles dans un langage unifié Modelica (chapitre 3 : II). Cette solution va s'avérer insuffisante et nécessite l'exploitation d'une solution complémentaire.

Une standardisation des composants de description des modèles (modèle seul ou le modèle avec son solveur de résolution) et de leurs interfaces de communication (chapitre 3 : III) est une solution qui s'avère pertinente pour compléter la première et ainsi parvenir à une simulation du système complet du bâtiment.

Chapitre 2 :

**La diversité des modèles rencontrés
dans le bâtiment et les difficultés de
modélisation : illustration sur le cas de
la plateforme PREDIS MHI.**

I. Introduction

Le bâtiment est un système complexe composé de multiples sous systèmes hétérogènes, issus de plusieurs domaines d'ingénierie (thermique, électrique, mécanique, aéraulique ...) pouvant être modélisés selon diverses approches et décrits avec des niveaux différents de finesse. Dans le chapitre 1, nous avons conclu sur la nécessité d'une modélisation globale tenant compte de tous les composants de ce système, pour assurer une gestion performante de tous ces flux énergétiques, dans un nouveau contexte de bâtiments basses consommations et intelligents. Dans ce chapitre, nous allons donc proposer d'étudier cet aspect d'hétérogénéité et de diversité à travers le bâtiment test de Prédís (plus précisément, PREDIS MHI).

Dans cette perspectives, nous allons procéder, après une description du bâtiment en terme d'objectifs, de caractéristiques et de composition (partie I), à la modélisation de son enveloppe thermique (partie II), de son système de ventilation et chauffage : VMC double flux (partie III) ainsi que ses usagers (partie IV). Plusieurs typologies de modèles seront établies et adoptées selon le domaine physique étudié, l'objet étudié (cf. l'acteur humain) et le niveau de précision requis, qui varie selon les objectifs d'utilisation et selon les données et informations disponibles.

La procédure de modélisation proposée et étudiée dans ce chapitre en application au cas du bâtiment Prédís, peut être généralisée et adoptée pour la modélisation d'autres systèmes de bâtiment de même type et dans les mêmes objectifs (chapitre 1, II.3.1). Ces systèmes sont généralement soumis aux mêmes contraintes et confrontées aux mêmes problèmes.

En effet, le bâtiment est un système non linéaire régit par un grand nombre de variables et soumis à plusieurs incertitudes. On peut parler des incertitudes relatives à la réalisation (remplacement des matériaux initiaux par d'autres équivalents, non suivi des plans...), à la non maîtrise de certains acteurs comme la météo et le comportement de l'occupant, et aussi à des difficultés de mesures (températures, pressions...) rendant la modélisation de ce système plus délicate. Ces aspects seront également traités tout au long de ce chapitre.

II. Description du bâtiment PREDIS MHI

II.1. Objectifs du bâtiment PREDIS MHI

II.1.1. Le centre PREDIS « Smart networks for energy »

PREDIS est un centre d'innovation (recherche), formation (enseignement) et valorisation (industrie) dans le secteur de l'énergie [PRED 2012]. Implanté sur 2500 m² de locaux de l'école ENSE3¹⁸ (école de l'énergie, l'eau et l'environnement), ce centre regroupe un ensemble de plateformes technologiques jouant le rôle de démonstrateurs de la gestion intelligente de l'énergie. Les axes de recherche autour de ces plateformes se répartissent en trois principaux thèmes énergétiques :

- La production décentralisée : une centrale de cogénération à base de micro turbine à gaz, une pile à combustible et deux ensembles de 10 panneaux photovoltaïques sont à disposition.
- Les réseaux du futur : un simulateur hybride temps réel, un réseau industriel local (échelle 1/10^{ème}), un réseau de distribution (échelle 1/1000^{ème}) et 5 autres y seront à terme et un système de conduite et de supervision (SCADA¹⁹) sont installés.
- L'usage : un habitat tertiaire réel « PREDIS MHI » (Monitoring Habitat Intelligent)

II.1.2. La plateforme PREDIS MHI

Ce démonstrateur a été mis en place pour répondre à des enjeux énergétiques fondamentaux relatifs aux « technologies développés pour les bâtiments » et afin de permettre l'étude et la compréhension des consommations et des usages, en particulier en lien avec l'électricité, dans les bâtiments basse consommation dans la perspective d'aller vers une gestion optimale de ce nouveau type de bâtiments.

Il a été ainsi conçu selon une démarche HQE [WEB_HQE] qui « vise à obtenir des bâtiments confortables, sains et plus respectueux de l'environnement » de manière à aboutir à un bâtiment conforme à la réglementation thermique en vigueur à l'époque (RT 2005). Les principales caractéristiques de la plateforme en terme d'efficacité énergétique et de confort, annoncées lors de la phase de conception, étaient :

-Une consommation en chauffage inférieure de 50kWhEP/m²/an. Chauffage assuré par une VMC double flux (échangeur rotatif de chaleur, batterie d'eau chaude ...).

-Un nombre d'heures de surchauffe annuelle inférieur à 42 h (nombre d'heures d'occupation quand la température dépasse les 28°C).

Cette plateforme a été équipée d'un système de gestion technique centralisée GTC ainsi que d'autres instrumentations intelligentes (système d'allumage auto réglable de l'éclairage...) illustrant les problématiques de gestion énergétique dans les bâtiments

¹⁸ENSE3: <http://ense3.grenoble-inp.fr/l-ecole/>

¹⁹ SCADA: Supervisory Control And Data Acquisition

intelligents pour pouvoir étudier les défaillances de ces systèmes et mieux concevoir leurs algorithmes de contrôle commande.

En plus des activités d'enseignement réalisées autour de ce bâtiment, plusieurs axes de recherches se sont développés comme :

- La problématique du matériel et de l'interfaçage logiciel : tester des nouveaux équipements domotiques, étudier des protocoles de communications, développer des IHM, tester des nouvelles solutions d'interopérabilités (webservices, OSGI...) [BEN 2010] [MIL 2010].
- Analyse des usages et de leurs impacts énergétiques : répartition des consommations, valorisation du confort, étude de l'efficacité et de l'efficience des équipements [CHEN 2012].
- Modélisation multi-physique, confrontation aux mesures et analyse de sensibilité : modélisation de l'enveloppe, des équipements et des phénomènes thermiques, électriques et aérauliques (projet SimintheC et Fiabilité).
- Développer et valider des stratégies de contrôle commande afin de gérer de façon optimale les flux énergétiques du bâtiment (projet MULTISOL).

II.2. Description architecturale

La plateforme PREDIS MHI est située au premier étage du bâtiment, PREDIS UHT (Figure 25), qui s'étend sur 300 m² de surface et est composé de deux étages. Le rez-de-chaussée contient des bureaux de recherche ainsi qu'un espace d'expérimentation en génie électrique. Le premier étage est constitué principalement d'une salle informatique de cours, un espace bureau dédié à la recherche et un local technique (Annexe 3). Un puit de lumière a été ajouté dans la salle informatique pour tirer profit de la lumière du jour et ainsi réduire la consommation due à l'éclairage artificiel.



Figure 25. Un aperçu du bâtiment PREDIS

Ce bâtiment tertiaire a été rénové et restructuré à partir d'un bâtiment existant à l'intérieur de la coque existante, en gardant sa façade extérieure et certains murs. Ce concept de rénovation intérieure est représentatif de la rénovation des bâtiments à façades historiques (ne devant pas être modifiées) puisque tel est le cas de la façade de l'école ENSE3, inscrite comme patrimoine historique.

Ce bâtiment présente ainsi la particularité d'être un « bâtiment dans un bâtiment », s'isolant de l'influence directe des conditions météorologiques extérieures (flux solaires, vent...). Cette isolation permet de réduire l'impact de ces facteurs externes sur le comportement du bâtiment pour mieux étudier l'influence des acteurs internes (occupants, équipements), l'objectif principal de la conception de ce bâtiment. Ces acteurs internes sont devenus, ainsi, plus pesants dans la balance énergétique de la plateforme.

En plus de cette première isolation avec l'extérieur, des matériaux isolants performants ont été utilisés pour permettre d'atteindre les objectifs en terme de chauffage. Une couche de 14 cm de ouate de cellulose a été posée sur un bardage bois qui entoure les parois intérieures et extérieures.

Dans la suite de ce chapitre, nous allons nous focaliser sur la modélisation et la simulation des systèmes du premier étage uniquement (noté par la suite PREDIS) et plus spécifiquement les deux zones «Salle Informatique» et «Espace Bureau» (Figure 26).



Figure 26. L'espace bureau et la salle informatique de la plateforme PREDIS MHI

II.3. Equipements

Afin de répondre aux exigences des bâtiments basses consommations et de satisfaire au cahier de charges, le premier étage a été équipé des systèmes intelligents et énergétiquement performants que nous allons détailler maintenant.

II.3.1. Une VMC double flux

Un système de ventilation mécanique contrôlée double flux (VMC DF) est installé afin d'assurer la ventilation et le renouvellement de l'air par l'apport d'air neuf extérieur.

Cette **ventilation** des locaux de PREDIS n'est pas faite naturellement, par tirage thermique exploitant le gradient de températures entre l'intérieur et l'extérieur [CONC 2002] ou par mise en dépression du bâtiment et par action du vent [RUS 2005]. Un dispositif **mécanique** de ventilation (deux moteurs électriques de puissance nominale de 750 W avec leurs ventilateurs) a été utilisé pour le forçage de la circulation d'air. Un fonctionnement commandé doit être plus avantageux que la ventilation naturelle qui rend le bâtiment plus sensible aux conditions extérieures et aux variations météorologiques (vent, température). Toutefois, il faut être rigoureux durant la construction car une infiltration parasite de 0,1 vol.h¹ peut diviser par deux l'efficacité d'un échangeur de chaleur [TROC 2009].

La vitesse de rotation des moteurs permet d'assurer un débit constant **contrôlé** et régulé automatiquement en fonction de l'occupation et des températures de consigne (amélioration possible: en fonction du taux de CO₂ dans la zone ventilée).

Les ventilateurs assurent le renouvellement d'air dans le deux sens de circulation selon le principe « **double flux** » (Figure 27): en extraction de l'air vicié et en soufflage de l'air neuf, à travers deux circuits aérauliques indépendants et séparés.

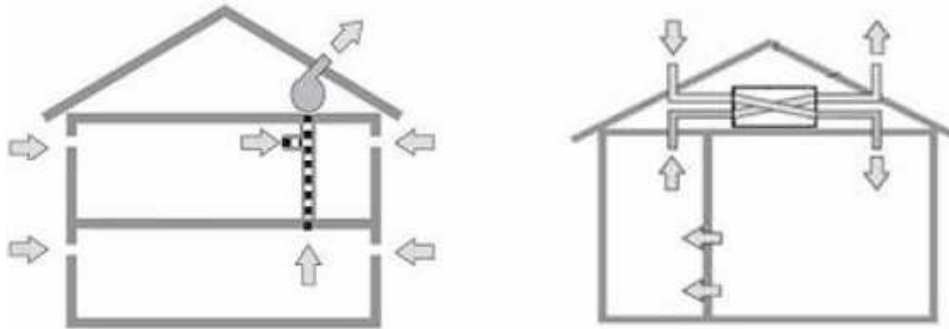


Figure 27. Le principe de la ventilation simple flux (gauche) double flux (droite) [LID 1966]

La technologie « double flux » présente l'avantage de pouvoir intercaler un échangeur de chaleur entre les deux flux d'air transférant la chaleur de l'air vicié (déjà chauffé par le système de chauffage ainsi que par les apports internes) à l'air neuf entrant, pour assurer son préchauffage. Un échangeur de chaleur de type rotatif a été installé dans PREDIS (Figure 28).

En raison de la séparation des deux flux d'air, on peut facilement procéder à un filtrage de l'air neuf avant sa distribution [KOF 2009] ainsi qu'à son chauffage, à l'aide d'une batterie d'eau chaude comme c'est le cas de l'installation de PREDIS.

Ce dispositif de VMC DF (Figure 28) minimise les risques de courant d'air froid, permet une meilleure maîtrise du débit d'air neuf insufflé et réduit considérablement les consommations énergétiques grâce à une récupération efficace de la chaleur (généralement supérieur à 80%) rendant le bâtiment plus performant comme prouvé dans plusieurs études et retours d'expériences [THIE 2008] [PREB 2007] [EFF 2008]...

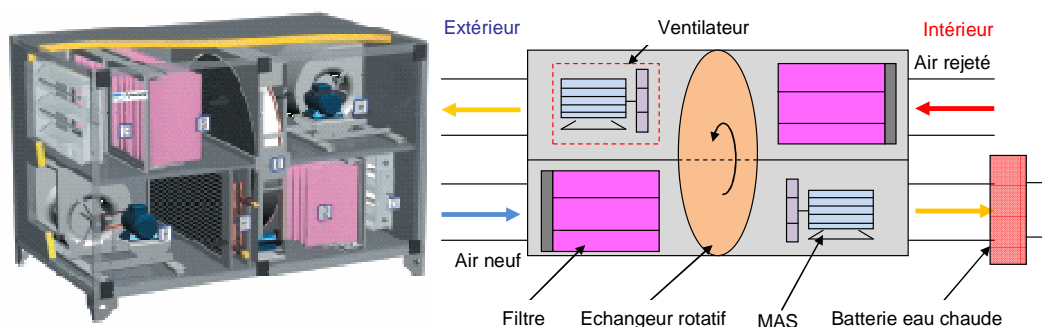


Figure 28. Composition de la VMC DF de PREDIS

II.3.2. Système de supervision et instrumentations expérimentales

Le bâtiment PREDIS MHI a été équipé d'une GTC constitué entre autres d'une centrale d'acquisition des données permettant à la fois de récupérer les mesures des différents

capteurs installés dans la plateforme mais aussi d'agir de manière optimale, selon les algorithmes de contrôle commande implémentés, sur les actionneurs (ouverture des vannes, vitesse de rotation des moteurs...) (Figure 29).

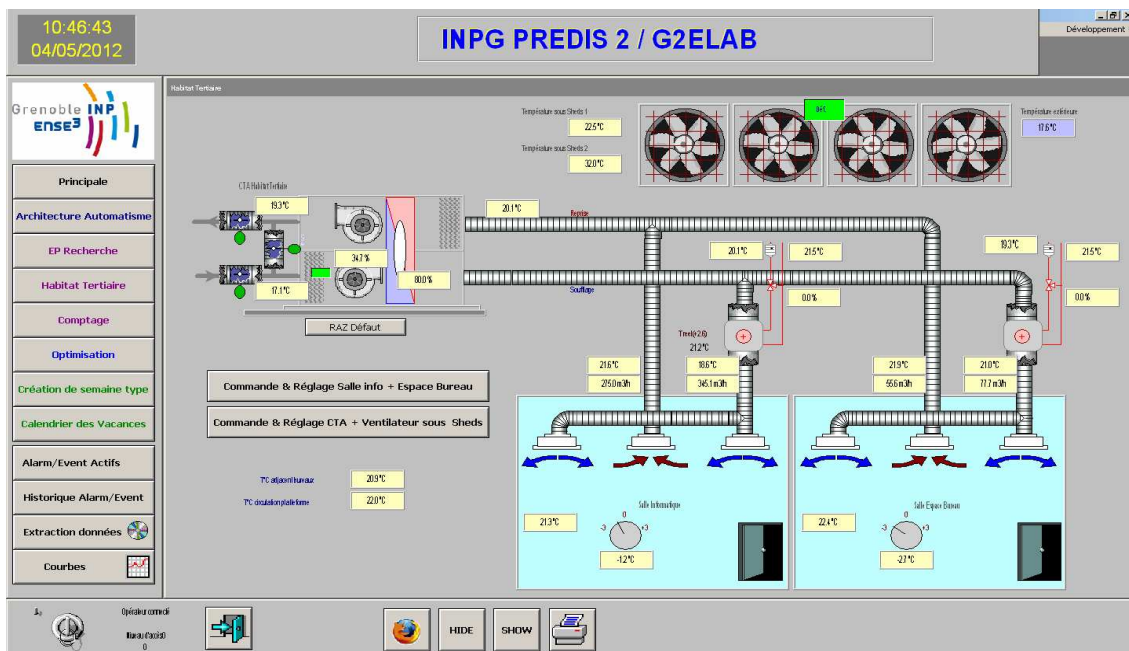


Figure 29. L'interface de la GTC de supervision du premier étage

Cette GTC permet à l'exploitant du bâtiment de surveiller le bon fonctionnement de ses différents systèmes (essentiellement la VMC DF en ventilation et en chauffage) à travers les différentes mesures affichées et de pouvoir en même temps agir sur les consignes de pilotage en cas de problème de dysfonctionnement. Toutes les mesures collectées ainsi que les consignes données sont sauvegardées et peuvent être extraites pour d'autres utilisations comme la validation des modèles et leurs confrontations aux mesures.

Afin de pouvoir mesurer les différents flux énergétiques et différentes informations utiles pour la régulation effectuée par le système de GTC, une instrumentation particulière s'appuyant sur diverses technologies de capteurs a été mise en place [MIL 2010] : compteurs pour la mesure des différents consommations électriques (éclairage, PC portables, moteurs électriques des ventilateurs...), capteurs de présence, des pressostats et essentiellement des débitmètres et des capteurs de températures qui seront exploités lors de la validation des modèles de la VMC DF. La figure ci-dessous (Figure 42) présente les principaux capteurs exploités lors de la modélisation des composants de la VMC.

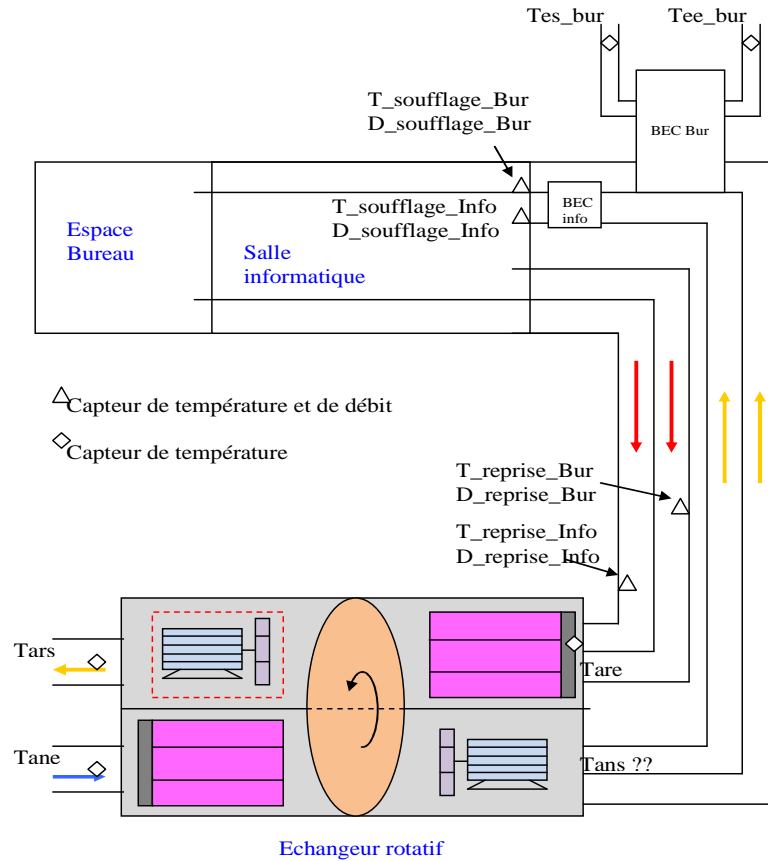


Figure 30. Les capteurs utilisés pour la modélisation de la VMC

II.3.3. Autres équipements

Pour, l'éclairage artificiel des zones, des tubes néons basses consommations ont été utilisés. Ils sont actionnés par un relais commandé par un détecteur de mouvement dans la zone correspondante. En plus, ils sont équipés d'un système d'obscurcissement « dimming system », permettant un autoréglage en fonction de la lumière du jour.

Les deux salles sont également équipées d'ordinateurs portables (pour la recherche et l'enseignement) moins consommateurs d'énergie que les ordinateurs fixes et disposant d'une capacité de stockage électrochimique intéressante pour la gestion énergétique.

III. Modélisation de l'enveloppe thermique

Deux principaux types de modèles peuvent être adoptés lors de la modélisation du comportement thermique d'un bâtiment [ZAY 2011]. Le premier s'intéresse à une modélisation fine avec l'étude des échanges thermiques et le second s'arrête à une modélisation simplifiée proposant des modèles d'ordre réduit. Les deux types de modèles sont unidimensionnels (1D) et visent l'étude de l'évolution des températures intérieures en fonction des excitations internes et externes.

Ces deux démarches seront appliquées au bâtiment PREDIS au travers, d'un côté, d'une modélisation analytique se basant sur des bilans thermiques aux niveaux des mailles de chaque zone, réalisés par le logiciel dédié COMFIE. D'un autre côté, on va utiliser un modèle empirique réduit, de type circuit électrique équivalent, dont les paramètres seront identifiés à partir des résultats de simulations du modèle COMFIE ou à partir des mesures.

III.1. Le modèle de l'enveloppe dans COMFIE Pleiades

III.1.1. Description et hypothèses du modèle

Dans le cadre de la thèse d'Hervé Chenailler [CHEN 2010], un modèle du bâtiment PREDIS MHI a été développé sous COMFIE Pleiades à partir du projet préétabli par le bureau d'étude l'ayant réalisé. Bien que le premier étage soit uniquement concerné par cette étude, le bâtiment a été modélisé dans sa globalité pour mieux étudier les interactions entre les différentes zones.

Plusieurs problèmes ont été rencontrés lors de la modélisation de ce bâtiment assez complexe (Figure 31) conduisant à l'adoption de plusieurs hypothèses :

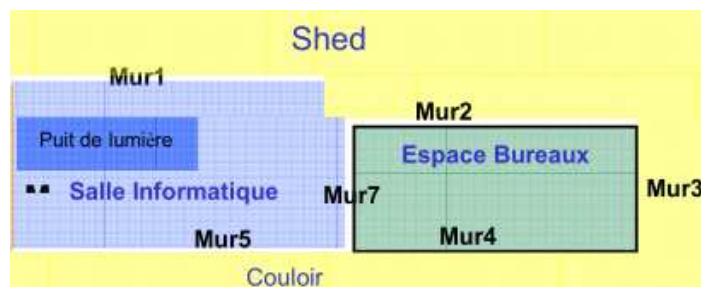


Figure 31 Vue de dessus des zones modélisées dans Alcyone (module de COMFIE Pleiades)

- Limitation de l'espace d'étude : pour représenter le principe du bâtiment à l'intérieur d'un autre bâtiment sans avoir à modéliser la totalité des locaux, une zone virtuelle « mitoyenne » a été introduite pour délimiter la zone d'étude et approximer les zones non étudiées. Cette zone mitoyenne impose une température T constante (19°C dans notre cas) à ses bords.

- Modélisation du shed : Le shed représente une « zone vide » (ne contenant que de l'air et sans équipement) mais qui s'étend sur l'équivalent de plusieurs étages. Etant donné que l'étude dans Alcyone se fait étage par étage par projection par rapport à un niveau donné, il était indispensable de représenter séparément la « zone vide » de chaque étage. Pour cela

des parois ont été ajoutées pour spécifier les délimitations horizontales. Un plancher très fin (plâtre 10 mm) a été utilisé pour séparer les étages du shed tout en réduisant son impact thermique.

- Modélisation des fenêtres internes et du puit de lumière : La notion de fenêtre dans COMFIE nécessite obligatoirement un contact avec l'extérieur. Or Prédic est un bâtiment dans un bâtiment, et les zones étudiées ne sont pas en contact direct avec l'extérieur ce qui bloque l'utilisation des fenêtres dans COMFIE. Pour y remédier, des portes ayant les mêmes dimensions que les fenêtres ainsi que les mêmes conductivités ont été utilisées pour la modélisation. La conductivité des matériaux est ainsi conservée mais pas leur degré d'opacité.

- Choix de la composition et des dimensions de certaines parois : Pour une question économique, certaines parois de l'ancien bâtiment ont été conservées. Leurs caractéristiques restent inconnues et étaient approximées dans COMFIE.

- Représenter le contact direct avec le sol : Sans vides sanitaires, le contact direct avec le sol est représenté par une surface de même température constante (estimé à 11° ici).

- Détermination des données météorologiques : Etant donné qu'aucun capteur de flux solaires (direct diffus et total) [BES 2000] n'est installé, les données météo de la station Chambéry ont été utilisées.

- Définition des scénarios: Même si plusieurs informations peuvent être récupérées depuis la GTC (puissance dissipée : en supposant 1W électrique consommé= 1W thermique dissipée, débit de ventilation...), l'information sur le nombre d'occupants n'est pas disponible ce qui nécessite son approximation.

Toutes ces hypothèses et approximations ainsi que d'autres problèmes (dysfonctionnement de la ventilation, calibrage des sondes, disponibilité des mesures sur des petits échantillons...) ont induit un écart non négligeable (2 degrés en moyenne) entre les résultats de simulation dans COMFIE et les mesures des températures [CHEN 2010].

III.1.2. Traitement du modèle par le cœur de calcul de COMFIE

La constitution du modèle global de l'enveloppe du bâtiment dans ce logiciel est menée tout d'abord pour chaque zone thermique indépendamment des autres zones en se basant sur les équations d'échange thermique établies pour chaque maille élémentaire de la zone étudiée (Annexe 4). Les parois sont donc décomposées en mailles élémentaires de manière à réduire au maximum leurs nombres afin d'accélérer les calculs tout en gardant homogènes les températures [ALG] [SAL 2005]. Le bilan thermique établi au niveau de chaque maille, conduit au système matriciel formalisé ainsi :

$$\begin{cases} C \cdot \dot{T} = A \cdot T + E \cdot U \\ Y = J \cdot T + G \cdot U \end{cases} \quad \text{Sys 1}$$

Ensuite, une linéarisation autour du point d'équilibre dynamique ($T=T_0-A^{-1}.E.U$) et un changement de base (base modale : $T = P.X$) vont permettre la diagonalisation des matrices. Les termes diagonaux représentent les différentes constantes de temps du système [DAUT 1997]. Cette technique d'analyse modale [MAR 1966] [BAC 1984] permet ainsi de réduire le

nombre de variables d'état en fonction de leur dynamique. En effet, certaines constantes de temps sont petites, et les termes du vecteur d'état atteignent très rapidement leur régime permanent. On peut alors les négliger dans le calcul en ne prenant en compte que les constantes de temps les plus significatives.

Suite à cette étape, les équations décrivant chaque zone sont couplées pour former un système global représentant la totalité du bâtiment et qui est de la forme [GAA 2011]:

$$\begin{cases} \dot{X} = F \cdot X + B^\alpha \cdot \dot{U} + B^\beta \cdot \dot{Y} \\ \dot{Y} = H \cdot X + S^\alpha \cdot U + S^\beta \cdot Y \end{cases} \quad \text{Sys 2}$$

Ce système peut alors être intégré sur un pas de temps Δt , ce qui permet d'obtenir les valeurs de X et Y au temps $(n+1) \cdot \Delta t$, notées X^{n+1} et Y^{n+1} , en fonction des valeurs X^n et Y^n au temps $n \cdot \Delta t$ (méthode récursive). On note de même U^{n+1} et U^n les vecteurs des sollicitations aux temps $(n+1) \cdot \Delta t$ et $n \cdot \Delta t$. Le système [2] devient après intégration :

$$\begin{cases} X^{n+1} = \exp(F \cdot \Delta t) \cdot X^n + W^\alpha \cdot (U^{n+1} - U^n) + W^\beta \cdot (Y^{n+1} - Y^n) \\ Y^{n+1} = MGIF \cdot X^n + MGIE \cdot U^{n+1} + MGID \cdot U^n - MGIC \cdot Y^n \end{cases} \quad \text{Sys 3}$$

$\exp(F \cdot \Delta t)$, W^α , W^β , $MGIF$, $MGIE$, $MGID$, $MGIC$: matrices du système, résultantes de l'intégration

La résolution de ce système à chaque pas de temps Δt , permet l'évaluation des températures intérieures des différentes zones, pour des scénarios prédéfinis d'apports internes.

Les matrices des systèmes d'états (Sys 2 et Sys 3), correspondant à la structure du bâtiment, ainsi que le vecteur des sollicitations U (extérieures : température extérieure et flux solaires incidents sur chaque parois et intérieures : puissances internes des occupants et des équipements), peuvent être exportées par COMFIE et sauvegardées sous la forme de fichiers texte dans son répertoire de travail temporaire.

III.2. Le modèle réduit de l'enveloppe

III.2.1. L'intérêt du modèle réduit

Bien que COMFIE assure une modélisation fine de l'enveloppe, la description d'un bâtiment y est lourde en terme de temps de calcul et d'effort de saisie. Elle est également complexe et nécessite la prise en compte de plusieurs paramètres et données qui ne sont pas nécessairement connus par l'utilisateur, comme c'est le cas de la conductivité thermique de certains murs préexistants, les charges internes ou le comportement des occupants... Ceci justifie le recours à plusieurs hypothèses pouvant affecter les précisions des résultats et réduire les avantages de l'utilisation d'un tel niveau de finesse assez élevé [HEN 2012] (cf chapitre 1). De plus, ce modèle fin présente l'inconvénient du coût de calcul élevé qui croît avec la complexité du modèle.

En outre, un modèle « COMFIE » est caractérisé par un ordre généralement élevé (plusieurs dizaines de variables d'état), malgré la réduction modale qui n'est effectuée qu'à partir de l'ordre 10 par zone. Or, dans certaines applications, comme le contrôle commande,

l'optimisation ou la prédiction..., un ordre faible des modèles et un court temps de calcul sont exigés.

Pour ces raisons, un autre type de modèle sera étudié. Il s'agit d'un modèle analytique basé sur le principe de circuit électrique équivalent, dont les paramètres peuvent être déterminés en lien direct avec la géométrie et les propriétés physiques du système, ou par identification.

III.2.2. Description du modèle réduit

Cette modélisation se repose sur une analogie thermique / électrique (Figure 32) permettant d'établir un circuit électrique équivalent simple pour une zone thermique du bâtiment (RC²⁰, R2C2²¹, R3C2) [ROUX 1984] [LE 2008]...

Flux	⇔	Courant électrique
Flux imposé	⇔	Générateur de courant
Température	⇔	Potentiel électrique
Température imposée	⇔	Générateur de tension
Conductance thermique	⇔	Conductance électrique
Capacité thermique	⇔	Condensateur électrique

Figure 32 L'analogie thermique électrique

Un modèle de type R3C1 a été adopté et adapté au cas d'étude de la salle informatique de la plateforme PREDIS MHI (Figure 33). Etant donné que les parois de la zone étudiée ne sont pas en contact direct avec l'extérieur (température extérieure), les températures des zones avoisinantes doivent être prises en compte dans le schéma équivalent. Tous les murs sont supposés avoir la même inertie (même C_m) et les flux solaires sont négligés.

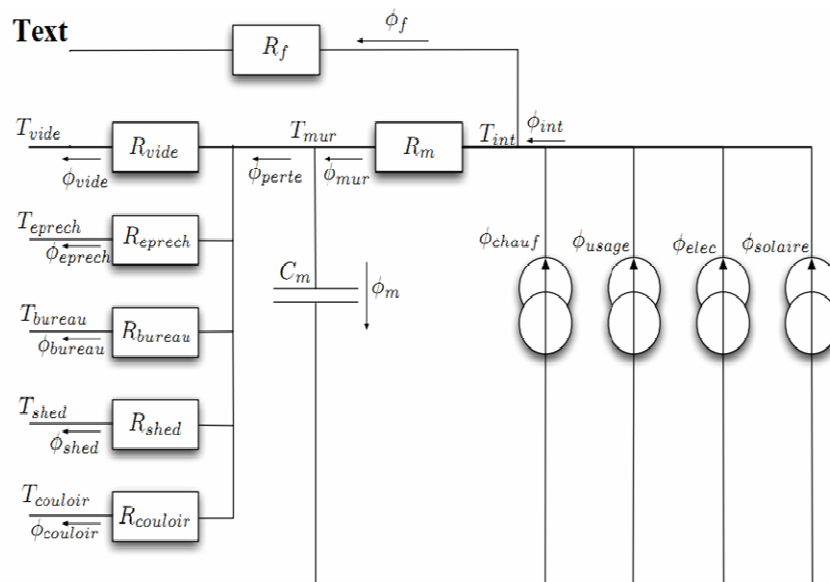


Figure 33 Le schéma électrique équivalent de l'enveloppe thermique

²⁰ RC : schéma électrique équivalent composé d'une résistance et d'une capacité

²¹ 2R2C : Il est ici composé de deux résistances et de deux capacités...

Ce modèle se traduit mathématiquement par le système d'équation suivant :

$$\frac{d}{dt} T_{mur} = \left[\frac{1}{C_m} \left(\frac{-1}{R_m} - \sum_{\text{Températures Voies} \sin es=1}^5 \frac{1}{R_i} + \frac{R_f}{R_m(R_m + R_f)} \right) \right] T_{mur}$$

$$+ \frac{1}{C_m} \left[\frac{1}{R_{vide}} \frac{R_f}{R_m + R_f} \frac{R_f}{R_m + R_f} \frac{R_f}{R_m + R_f} \frac{R_f}{R_m + R_f} \frac{R_f}{R_m + R_f} \frac{1}{R_{eprech}} \frac{1}{R_{shed}} \frac{1}{R_{bureau}} \frac{1}{R_{couloir}} \frac{1}{R_m + R_f} \right] U$$

$$T_{int} = \left[\frac{R_f}{R_m + R_f} \right] T_{mur} + \frac{R_m R_f}{R_m + R_f} \left[\begin{matrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \frac{1}{R_f} \end{matrix} \right] U$$

avec $U = \left[T_{vide} \quad \phi_{chauf} \quad \phi_{usage} \quad \phi_{elec} \quad \phi_{solaire} \quad T_{eprech} \quad T_{shed} \quad T_{bureau} \quad T_{couloir} \quad T_{ext} \right]$

Il s'agit ensuite d'identifier les paramètres de ce modèle à partir des résultats de COMFIE pour les mêmes scénarios d'usage.

III.2.3. Procédure d'identification des paramètres du modèle

L'idée est désormais de faire varier les paramètres du modèle de manière à faire coïncider, au mieux, la température ambiante calculée par ce modèle avec la température simulée par COMFIE (la référence) [CAV 2011]. Cette variation est réalisée automatiquement par un programme d'identification utilisant un algorithme d'optimisation déterministe de type quasi newton [SHA 1970] (ici « fmincon » de Matlab). Cette optimisation permet de déterminer le jeu de paramètres optimal permettant de minimiser la fonction objective : l'écart entre les deux températures (Figure 34).

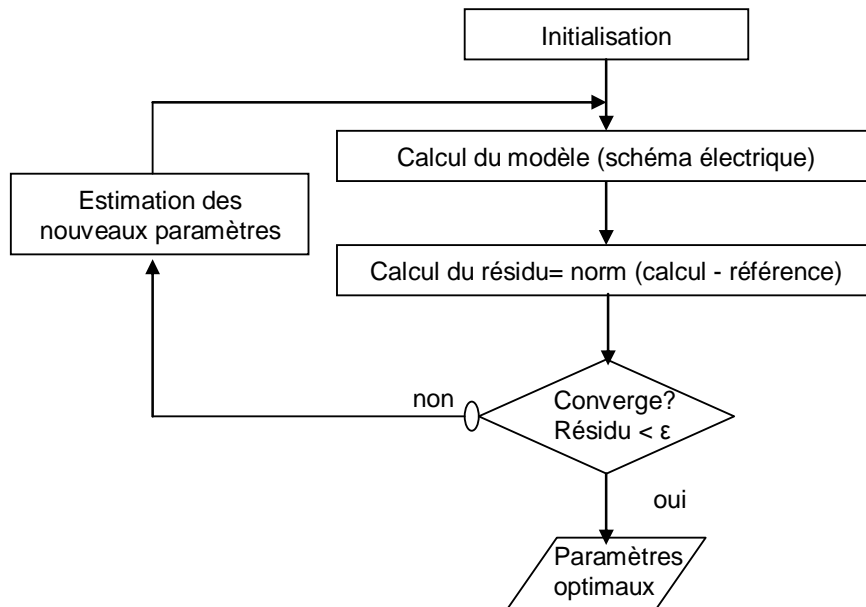


Figure 34 L'algorithme d'identification

Pour assurer une identification performante des paramètres de ce modèle, il est indispensable d'effectuer un choix judicieux des valeurs initiales. En effet, l'algorithme d'optimisation employé est un algorithme déterministe très dépendant des valeurs initiales et il peut facilement se bloquer sur des minima locaux [COU 2002]. Des méthodes stochastiques

peuvent être utilisées mais sont très coûteuses en temps de calcul en raison d'une convergence très lente. Par contre, les paramètres du schéma équivalent étant directement liés à des grandeurs physiques, nous pouvons calculer des valeurs initiales approximatives. Les résistances équivalentes et les capacités initiales ont donc été calculées à partir des données sur les parois: $C_m=0.18$, $R_m=2.8 \cdot 10^{-4}$, $R_f=2.2 \cdot 10^{-3}$, $R_{vide}=4.27$, $R_{eprech}=1.04 \cdot 10^{-3}$, $R_{shed}=16.14$, $R_{bureau}=6.45$, $R_{couloir}=0.04$. Avec ces valeurs la température calculée par le modèle est assez éloignée de celle simulée par COMFIE (Figure 35).

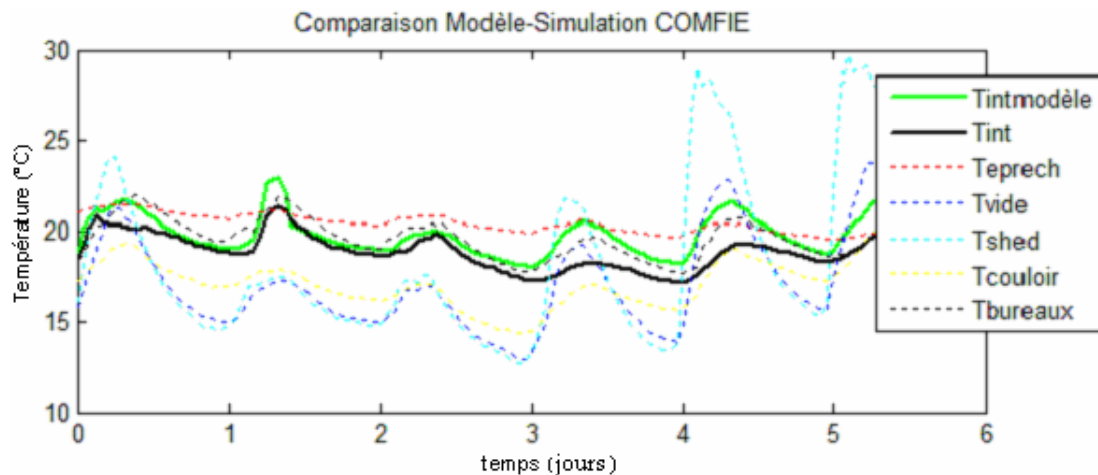


Figure 35 Comparaison entre le modèle avec les paramètres initiaux et la simulation

Les nouveaux paramètres résultants de l'optimisation finale: $C_m=5.6$, $R_m=7.02 \cdot 10^{-6}$, $R_f=0.09$, $R_{vide}=170.72$, $R_{eprech}=1.3 \cdot 10^{-3}$, $R_{shed}=654.71$, $R_{bureau}=0.16$, $R_{couloir}=9.92 \cdot 10^{-4}$, ont permis l'amélioration des résultats précédents (Figure 36) et ont aboutit à une erreur maximale de 1 degré.

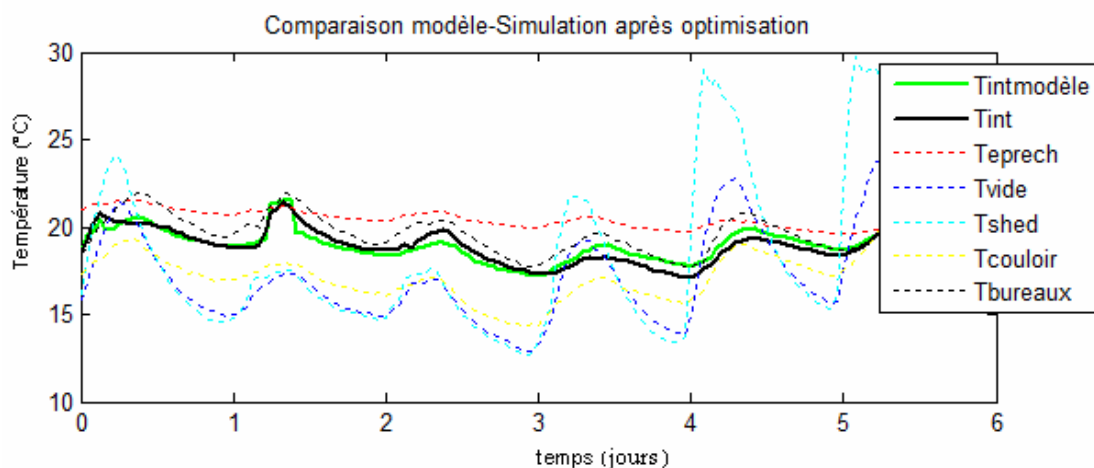


Figure 36 Comparaison entre le modèle avec les paramètres optimisés et la simulation

Afin de vérifier la robustesse de ces paramètres et juger la qualité de prédiction du modèle identifié, une validation croisée sur un autre échantillon de mesures a été réalisée. Il s'agit de valider si les résultats calculés par le modèle collent avec les résultats de COMFIE sur une période différente de la période d'identification. Les résultats obtenus ne sont pas toujours satisfaisants essentiellement lors des grandes variations des sources (température

extérieure, débit d'air, occupations, usage). Des travaux de recherche sont en cours afin de repérer les sources de problèmes et d'incertitudes pour améliorer les capacités de prédiction de ce modèle réduit.

III.3. Conclusion

On note ainsi que la nature du modèle et son niveau de finesse peut varier selon les objectifs étudiés : un modèle numérique tel que COMFIE pour la simulation détaillée de l'enveloppe, et un modèle réduit pour une prédiction comportementale rapide.

L'étude de ces deux types de modèles a mis l'accent sur une difficulté majeure dans le processus de modélisation. Il s'agit de la détermination des scénarios des puissances internes : équipements et usagers qui sont très influents sur la variation des résultats. D'où le besoin d'une modélisation plus fine des ces acteurs.

IV. Modélisation des équipements : VMC double flux

IV.1. La partie électrique : Moteur et variateur de vitesse

Cette partie est composée principalement de deux machines asynchrones identiques ($P_n=750$ W, $\Omega_n=2855$ tr/min, $\eta_n=0,73$) commandées chacune par un variateur de vitesse ($f_n=12$ kHz). Une partie de commande envoie des signaux aux interrupteurs de son circuit d'électronique de puissance (onduleur MLI) afin de changer la tension et la fréquence d'entrée du moteur agissant ainsi sur la vitesse des ventilateurs et donc sur le débit d'air.

Le but principal de la modélisation de cette partie, réalisée dans le cadre du master de Hoang Anh Dang [DANG 2010] pour lequel j'ai participé à l'encadrement, est de construire un modèle simplifié de ce système. L'objectif étant d'évaluer la consommation énergétique et la vitesse de rotation des moteurs qui impose le débit d'air forcé. Etant donné que la constante de temps de l'enveloppe thermique est plus grande que les constantes de temps des systèmes électriques, leurs régimes transitoires seront négligés et seulement les régimes permanents seront étudiés. Ainsi, le modèle de la partie électrique possède :

- 4 entrées : tension, fréquence, consigne de vitesse de rotation et couple de résistance ;
- 3 sorties : vitesse de rotation du moteur, rendement du moteur et rendement du circuit EP.

Vu la dépendance entre le circuit de commande et le moteur et pour des facilités de modélisation, les deux parties du variateur de vitesse (partie commande et partie électronique de puissance) ont été séparées lors de la modélisation (Figure 37).

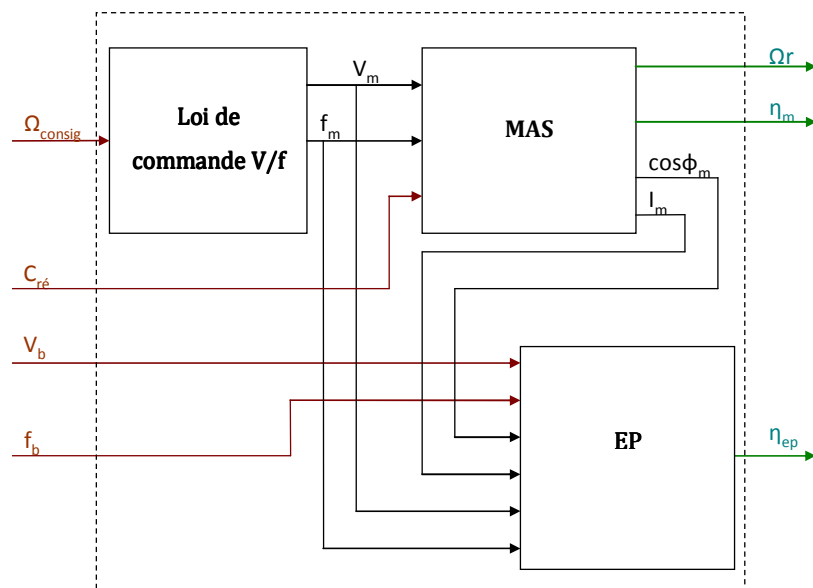


Figure 37 Modèle général des éléments de la partie électrique

IV.1.1. La machine asynchrone

Le modèle de la MAS se base sur un schéma équivalent qui se compose de résistances et réactances linéaires (Figure 38). Ce schéma électrique classique permet d'étudier le fonctionnement du moteur en régime permanent et quelques phénomènes en régime spécial

(démarrage, rotor bloqué, à vide) [SEG 1982] [RAZ 2006]. Il s'agit donc d'un modèle analytique dont on peut facilement identifier les paramètres grâce aux données constructeurs.

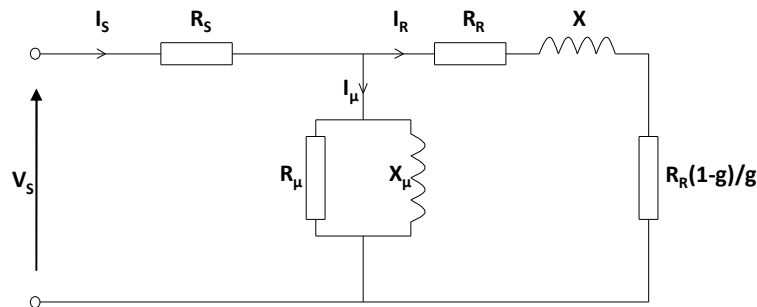


Figure 38. Schéma équivalent du moteur asynchrone

A l'issue de cette identification réalisée sous Mathcad [DANG 2010], les paramètres du circuit équivalent résultants sont regroupés dans le tableau suivant (Tableau 3) :

R_S (Ω)	R_R (Ω)	X (H)	R_μ (Ω)	X_μ (H)
13,909	7,106	0,041	$1,709 \times 10^4$	0,795

Tableau 3. Les valeurs des paramètres identifiées du circuit équivalent

Grâce à ces paramètres, aux données constructeurs et aux expressions analytiques caractérisant le fonctionnement de ce type de machine, les performances (rendement) et les pertes du moteur peuvent être calculées (Annexe 5).

IV.1.2. Le variateur de vitesse

Concernant les performances du variateur de vitesse, les pertes de la partie commande seront négligées parce qu'elles sont très faibles (1/1000 des pertes globales) [DANG 2011]. La partie commande sera étudiée pour déterminer la tension et la fréquence modulées transmises aux autres parties et nécessaires pour leurs calculs.

IV.1.2.1. Les pertes du circuit d'électronique de puissance

Ce circuit est composé, comme présenté dans la figure (Figure 39), principalement de :

- un redresseur qui, connecté à une alimentation alternative triphasée du réseau, génère une tension continue à ondulation résiduelle.

- un circuit intermédiaire jouant le rôle du «lissage» de la tension de sortie du redresseur afin d'améliorer sa composante continue.

- un onduleur qui génère le nouveau signal alternatif à tension et/ou fréquence variables sur lequel vient se connecter la MAS.

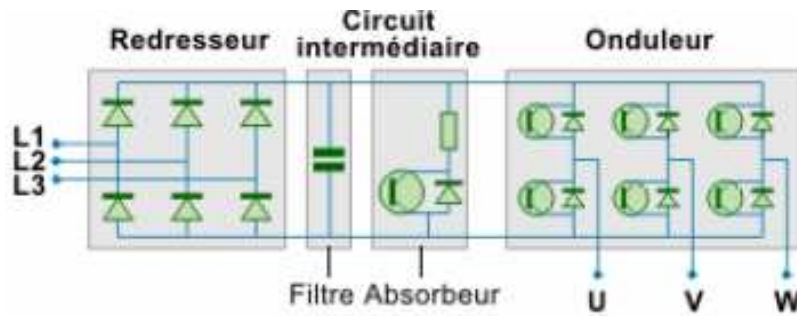


Figure 39. Composition du circuit d'électronique de puissance du variateur de vitesse

Plusieurs hypothèses ont été adoptées en vue d'assurer la modélisation du circuit EP étant donné que l'objectif de cette modélisation est le calcul estimatif de ses pertes sans chercher à déterminer son fonctionnement précis. La principale hypothèse consiste à considérer que les pertes du redresseur et du circuit intermédiaire seront négligées devant celle de l'onduleur vu la fréquence de hachage élevée de ce dernier (facteur 40 : 1,8 W par rapport à 70 W dans l'onduleur).

L'onduleur considéré est commandé en Modulation de Largeur d'Impulsion (MLI) de type intersective [REG 2003]. Les pertes dans ce circuit correspondent aux pertes de ces 6 interrupteurs qui se composent chacun d'un IGBT et d'une diode anti-parallelèle. Les pertes de ces composants sont séparées en 2 types de pertes : pertes en conduction et pertes en commutation [DEL 2004].

Les pertes en conduction sont des pertes joules dues aux résistances internes des composants électroniques imparfaits, à l'état passant lors qu'ils sont traversés par un courant :

$$\text{Pertes de l'IGBT : } \Delta P_{IGBT_cond} = V_{CE0} I_{IGBTmoy} + R_{IGBT} I_{IGBTeff}^2$$

$$\text{Pertes des diodes antiparallèles : } \Delta P_{d_cond} = V_{d0} I_{d moy} + R_d I_{d eff}^2$$

Les pertes en commutation sont expliquées par le fait que la coupure du courant n'est pas instantanée lors du passage d'un état à un autre en ouverture et en fermeture. Dans cette période de recouvrement, un courant et une tension sont détectés au niveau de chaque composant et qui y génèrent des pertes. Ces pertes sont déterminées à partir du calcul des énergies de commutations multipliées par la fréquence :

$$\text{Energies de commutation de l'IGBT : } W_{on} = \frac{W_{on\ ref}}{E_{ref} \cdot I_{ref}} \cdot E \cdot I \text{ et } W_{off} = \frac{W_{off\ ref}}{E_{ref} \cdot I_{ref}} \cdot E \cdot I$$

$$\text{Energies de commutation de la diode : } W_{d_com} \approx W_{doff_com} = I_d \frac{Q_{rr_ref}}{I_{d_ref}} E$$

IV.1.2.2. La loi de commande

Une loi de commande en V/f a été implémentée dans le variateur de vitesse. Cette loi est adaptée aux applications de types habitat (ventilateurs, pompes...), se contentant d'une commande simple qui ne nécessite pas ni une grande précision ni une action rapide

(contrairement à une commande vectorielle). Une commande scalaire en V/f variable est utilisée dans le variateur, adaptée à une charge aéraluque dont le couple varie en fonction de la vitesse (contrairement à une commande V/f constant).

Afin d'identifier cette loi de commande, plusieurs mesures de la tension en fonction de la fréquence ont été réalisées (Figure 40). Il s'agit d'une loi parabolique, donc pouvant être décrite sous la forme : $V = a.f^b + c$. L'identification de cette relation à partir des mesures aboutit à la loi de commande $V = 0,662.f^{1,475} + 12,938$, qui sera utilisée dans notre modèle.

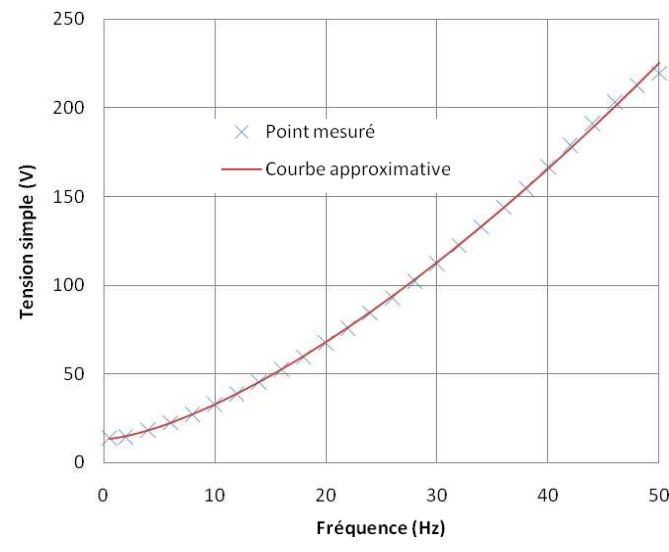


Figure 40. Fonctionnement en V/f couple variable

IV.2. Les échangeurs de chaleur

IV.2.1. Le principe de modélisation

Un échangeur de chaleur est un système dont le principe de fonctionnement consiste à transférer la chaleur d'un fluide chaud (indice 2) à un fluide froid (indice 1) sans contact direct entre eux à travers une paroi, souvent métallique pour favoriser les échanges. La quantité de chaleur transférée dépend des variables d'entrée des fluides (débits et températures) et de plusieurs paramètres comme :

- La configuration géométrique de l'échangeur : les surfaces d'échange...
- la nature des deux fluides et leurs caractéristiques thermiques : fluides identiques ou non : air/eau ; air/air ou eau/eau.
- Le régime de circulation des fluides [JAN 2009] (Figure 41): flux parallèles, contre courants ou courant croisés, fluides brassés ou non brassés.

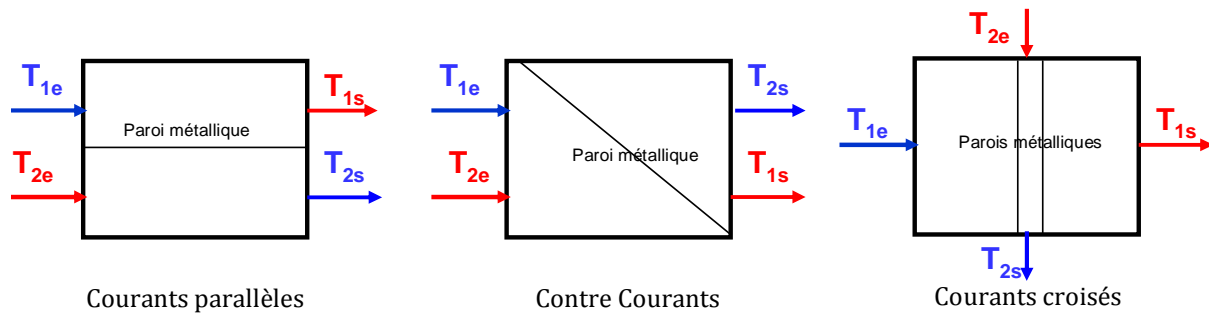


Figure 41. Les régimes de circulation de l'air dans les échangeurs de chaleur

La modélisation des échangeurs de chaleurs installés dans PREDIS (Echangeur rotatif et batterie eau chaude) doit permettre, tout en tenant compte de ces paramètres, d'étudier leurs performances énergétiques et de pouvoir déterminer les transferts de chaleur entre les deux fluides ainsi que les températures de fluides à la sortie. Pour modéliser ces deux échangeurs une méthode NUT- ϵ sera adoptée [MAC 1972] [MOR 1999] [MAR 2002a] [WET 1999a] [WET 1999b] [MAR 2002b]. Elle consiste à déterminer, en se basant sur cette relation, les caractéristiques nominales (coefficients d'échange) au point de fonctionnement nominal pour ensuite étendre ces calculs vers d'autres points. Seuls les transferts de chaleur seront considérés, les transferts de masse seront ainsi négligés.

Le calcul se base sur trois principales étapes :

Calcul des coefficients d'échange au point de fonctionnement nominal

Connaissant les données nominales du modèles au point de fonctionnement (fournies par le constructeur, diagrammes, mesures...) : l'efficacité et le débit correspondant, ainsi que sa configuration géométrique, on va déterminer les coefficients d'échange nominaux :

L'efficacité thermique d'un échangeur correspond au rapport entre le flux de chaleur effectivement transféré et le flux de chaleur maximal pouvant être transféré dans le cas d'un échangeur parfait (tube de longueur infinie avec un régime à contre courant) : $\epsilon = \frac{Q}{Q_{\max}}$.

L'efficacité peut s'exprimer en fonction de caractéristiques physiques de l'échangeur, des capacités calorifiques des fluides et dépend du régime d'écoulement :

$$\epsilon = f(NUT, C = \frac{C_{\min}}{C_{\max}}, \text{régime}) \quad (\text{Annexe 6})$$

Avec les capacités calorifiques minimale $C_{\min} = \min(C_1, C_2)$ et maximale

$$C_{\max} = \max(C_1, C_2) : C_i = C_{pi} \dot{m}_i$$

Ainsi le nombre d'unités de transfert nominal peut être exprimé en fonction de l'efficacité au point nominal : $NUT^{nom} = g(\epsilon^{nom})$

$$\text{D'un autre côté } NUT = \frac{UA}{C_{\min}} \text{ avec } \frac{1}{UA} \approx \frac{1}{hA_1} + \frac{1}{hA_2}$$

Donc le NUT nominal peut s'écrire également :
$$NUT = \frac{1}{C_{\min}} \left(\frac{1}{hA_1} + \frac{1}{hA_2} \right)^{-1}$$

On définit un coefficient déterminé selon la géométrie de l'échangeur $r = \frac{hA_1}{hA_2}$

A : surface d'échange et h : coefficient de transfert convectif de chaleur. Par exemple r est égal à 1 lorsque les deux fluides sont identiques et traversent la même surface d'échange.

Ainsi les deux coefficients d'échanges au point de fonctionnement hA_1^{nom} et hA_2^{nom} peuvent être calculés.

Calcul des coefficients d'échange autour du point de fonctionnement nominal:

En assimilant les circuits d'air à des tubes, la corrélation de Dittus-Boelter [INC 1996] peut être appliquée pour relier les coefficient d'échange au point de fonctionnement à celles

autour de ce point :
$$\frac{hA_i}{hA_i^{\text{nom}}} = k_i \left(\frac{\dot{m}_i}{\dot{m}_i^{\text{nom}}} \right)^\alpha$$
 : α est fixé selon la nature de deux fluides utilisés.

k_i : facteur de variation des propriétés du fluide à une température donnée. Cette variation sera négligée dans la suite (~ 1).

Calcul des performances de l'échangeur:

On peut maintenant procéder au calcul des performances de l'échangeur (efficacité et puissance échangée) ainsi que ses températures de sortie en se basant sur les mêmes équations utilisées précédemment mais de manière inverse :

- Connaissant les coefficient d'échange hA_1 et hA_2 , on peut déterminer NUT utilisant l'équation [3].
- L'efficacité de l'échangeur peut être ensuite calculée à partir du NUT avec la relation [2].

D'un autre côté, L'efficacité peut être exprimée en fonction des différences de températures de deux fluides à l'entrée et à la sortie de l'échangeur :

En effet l'efficacité $\mathcal{E} = \frac{Q}{Q_{\max}}$. Or $Q = C_1(T_{1e} - T_{1s}) = C_2(T_{2e} - T_{2s})$ et $Q_{\max} = C_{\min}(T_{2e} - T_{1e})$

On aboutit donc aux expressions suivantes qui permettent de déduire les températures

de sortie :
$$\mathcal{E} = \frac{Q}{Q_{\max}} = \frac{C_1(T_{1e} - T_{1s})}{C_{\min}(T_{2e} - T_{1e})} = \frac{C_1(T_{2e} - T_{2s})}{C_{\min}(T_{2e} - T_{1e})}$$

IV.2.2. La procédure de validation des modèles.

Une fois les équations du modèle des échangeurs établis, l'étape suivante consiste à identifier ses paramètres à partir des données du constructeur (masse de la roue, sa vitesse

nominale, débit nominal, régime d'écoulement), ou également à partir des mesures (efficacité nominale). Les données nominales peuvent être mesurées pour avoir des données plus adaptées au système installé car les équipements réels peuvent être mal dimensionnés (fonctionner en sous régime comme c'est le cas de la VMC de PREDIS) ou dégradé avec le temps.

On peut ensuite procéder à la validation des modèles par confrontation avec les mesures de la GTC (Figure 42). Il s'agit de comparer les variations des températures à la sortie de l'échangeur entre les mesures et le calcul effectué par les modèles avec les données expérimentales (débit, températures entrantes).

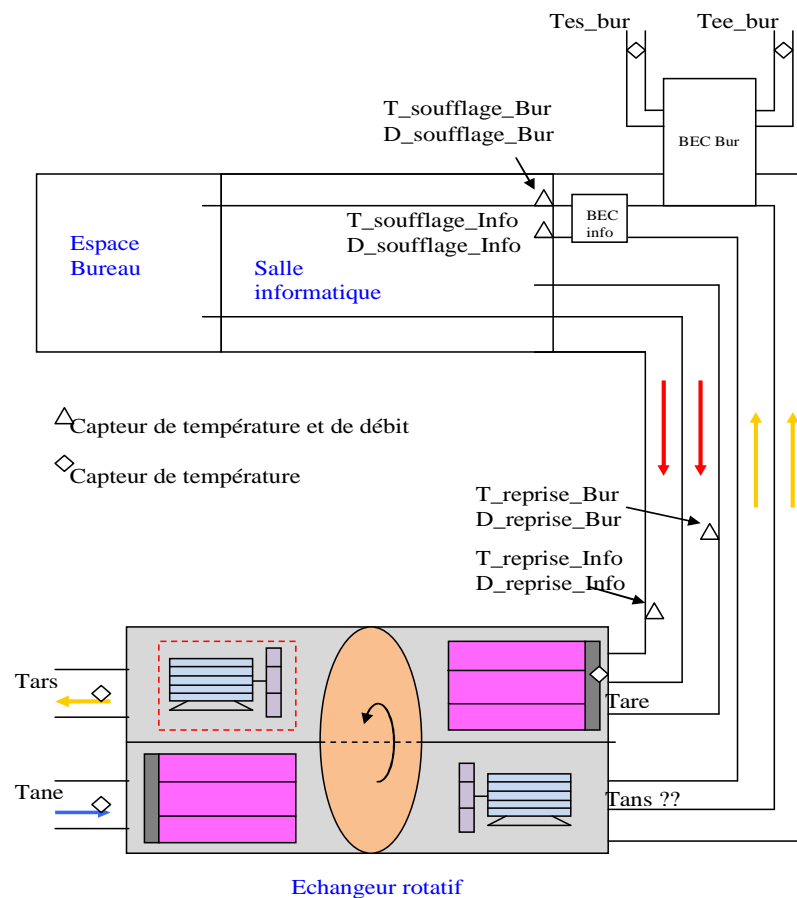


Figure 42. Les capteurs utilisés pour la modélisation de la VMC

Il est important de commencer par sélectionner une période valide de mesure (VMC fonctionne, GTC fonctionne) et qui peut répondre à nos objectifs essentiellement avec l'absence de certains capteurs. Par exemple, en l'absence d'une mesure directe à la sortie de l'échangeur rotatif, la température de soufflage sera exploitée. Il en résulte que pour pouvoir valider le modèle de l'échangeur seul, il faut choisir une zone de mesure où la BEC est éteinte (batterie d'eau chaude assurant le chauffage de l'air soufflé par la VMC).

Il est aussi indispensable de vérifier la validité de certaines mesures influentes sur la procédure de validation. Par exemple, en observant les températures de l'air neuf à l'entrée et

à la sortie de la VMC en l'absence de tout échange de chaleur, un écart important a été détecté et qui n'est pas constant tout au long de la période testée (Figure 43).

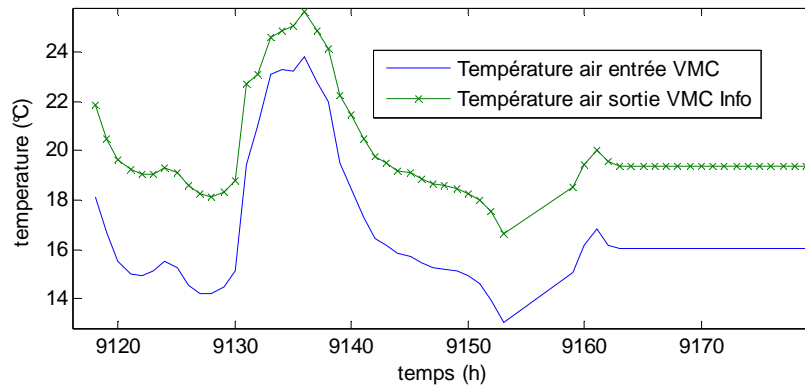


Figure 43. Différence de mesure des températures entre l'entrée (capteur Tane) et la sortie de la VMC (mesurées au niveau du soufflage dans la salle informatique: capteur T-soufflage_Info)

Il peut être dû aux échanges de chaleur avec les locaux traversés au niveau du réseau aéraulique (isolation imparfaite) comme il peut dériver d'un problème d'étalonnage des capteurs. La dernière hypothèse a été retenue en supposant négligeable les déperditions du circuit aéraulique. Pour remédier à ce problème des termes correctifs ont été introduits aux mesures :

$$T_{1e}^{corrigé} = a_1 T_{1e}^{mesure} + b_1 \text{ et } T_{1s}^{corrigé} = a_2 T_{1s}^{mesure} + b_2$$

En se basant sur la même procédure exposée dans le paragraphe concernant le modèle réduit de l'enveloppe, une identification a été effectuée afin de déterminer les 4 paramètres de manière à minimiser l'erreur entre T1 et T2 (Figure 44 gauche) ($a_1=0.9$ $b_1=3.4$, $a_2=1.1$, $b_2=-3.4$). Ensuite, et dans le but de vérifier la robustesse de ces paramètres, d'autres plages de mesures ont été observées pour vérifier que l'écart est toujours minimal (Figure 44 droite).

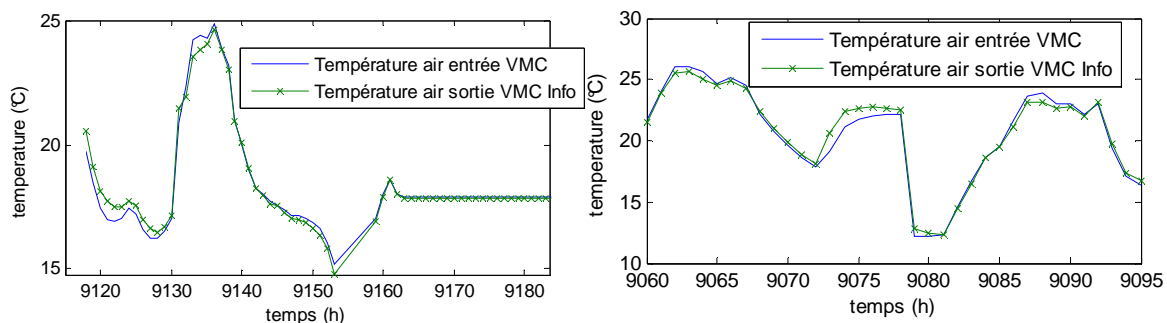


Figure 44. Les mesures de températures corrigées

IV.2.3. L'échangeur rotatif

IV.2.3.1. Caractéristiques du modèle

L'échangeur rotatif utilisé dans PREDIS est une roue en aluminium en forme de nid d'abeilles (Figure 45). Ce matériau accumulateur permet de récupérer la chaleur de l'air extrait vicié pour la céder ensuite au flux d'air neuf dès qu'il entre en contact avec. Cette matrice de masse M et de chaleur spécifique C_m est entraînée à vitesse variable N par un

moteur à courant continu dont la puissance consommée sera négligée ($P_n = 40 \text{ W}$) devant les puissances mises en jeu.

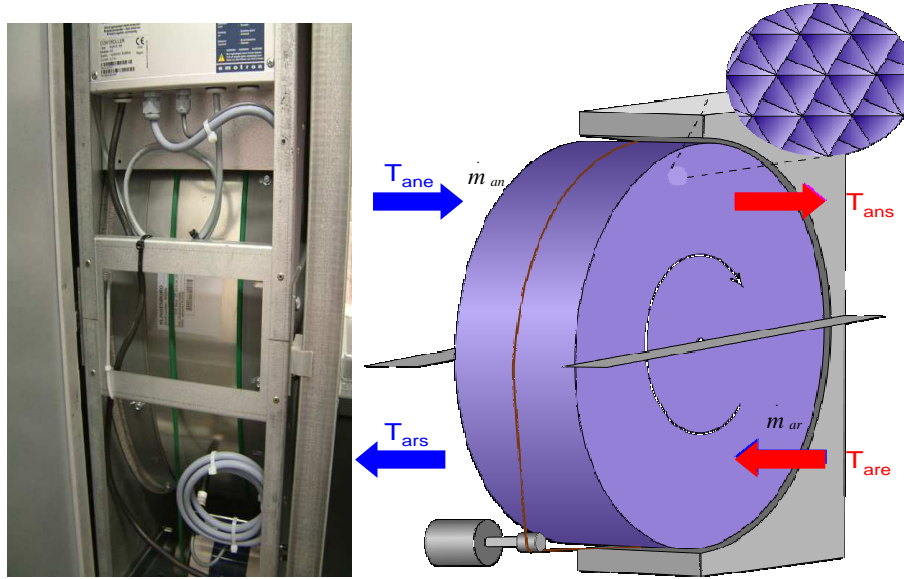


Figure 45. L'échangeur rotatif (image réelle : gauche et schéma : droite)

Pour la modélisation de cette roue, Kays [KAY 1984] propose une corrélation empirique reliant l'efficacité des échangeurs rotatifs à celle des échangeurs statiques à contre-courant et qui tient compte de cette rotation :

$$\varepsilon = \varepsilon_{cc} \left(1 - \frac{1}{9(C_r^*)^{1.93}} \right)$$

$$\text{avec le terme correctif } C_r^* = \frac{C_r}{C_{\min}} = \frac{M \cdot c_m \cdot N}{C_{\min}}$$

En appliquant l'hypothèse d'un échangeur à contre courant au point de fonctionnement ($C=1$ car même fluide : air-air, et même débit), la relation

$$NUT_{cc}^{nom} = \frac{\varepsilon_{cc}^{nom}}{1 - \varepsilon_{cc}^{nom}} \text{ peut être utilisée. Ainsi } NUT_{cc}^{nom} = \frac{\varepsilon^{nom} \left(1 - \frac{1}{9(C_r^*)^{1.93}} \right)^{-1}}{1 - \varepsilon^{nom} \left(1 - \frac{1}{9(C_r^*)^{1.93}} \right)^{-1}}$$

Etant donné que les deux surfaces de la roue en contact avec l'air neuf et avec l'air rejeté sont identiques, le terme $r = \frac{hA_{an}}{hA_{ar}} = 1$. hA_{an}^{nom} et hA_{ar}^{nom} peut être ainsi calculé comme décrit précédemment.

En dehors de ce point de fonctionnement les relations $\frac{hA_{an}}{hA_{an}^{nom}} = \left(\frac{m_{an}}{m_{an}^{nom}} \right)^{0.78}$ et

$\frac{hA_{ar}}{hA_{ar}^{nom}} = \left(\frac{m_{ar}}{m_{ar}^{nom}} \right)^{0.78}$ [WET 1999a] sont appliquées pour permettre le calcul de l'efficacité :

$\varepsilon_{cc} = \frac{1 - \exp[-NUT_{cc} \cdot (1 - C)]}{1 - C \exp[-NUT_{cc} \cdot (1 - C)]}$ et déduire par la suite les températures de sortie et les quantités de chaleurs échangées.

IV.2.3.2. Implémentation et validation du modèle

Le modèle de l'échangeur rotatif (0D) a été implémenté dans Matlab. En plus de la détermination des valeurs nominales et de l'identification de ces paramètres, l'alimentation de ce modèle a nécessité l'extraction de plusieurs mesures et informations depuis la GTC (Figure 42). Il s'agit des entrées du modèles qui sont : les températures et les débits de l'air neuf (capteurs : *Tane* et *D_soufflage_Bur+D_soufflage_Info*), les températures et les débits de l'air rejeté à l'entrée de l'échangeur (capteurs : *Tare* et *D_reprise_Bur+D_reprise_Info*) ainsi que la vitesse de rotation de la roue déterminée à partir de la consigne envoyée par la GTC (Consigne roue). L'information sur la consigne de la vanne d'eau (Consigne vanne) était nécessaire pour sélectionner la période de validation (non fonctionnement de la BEC).

Après détermination des paramètres géométriques du modèle et identification des données nominales à partir d'un point de mesure stable, une comparaison est effectuée entre la température de l'air neuf à la sortie de l'échangeur calculée par le modèle (*Tans*: courbe rose) et celle mesurée par les capteurs au soufflage (*T_soufflage_info*: courbes rouge) (Figure 46).

On peut observer que les résultats calculés par le modèle suivent l'allure des mesures avec un réchauffement de l'air neuf vicié (*Tare*: courbe verte) en période de froid et « récupération de froid » (extraire de la chaleur de l'air neuf entré pour la transmettre à l'air sortant et ainsi refroidir l'air soufflé) en période de chaud.

Une faible erreur est obtenue ($\sim 2,5$ % en moyenne pour la période testée).

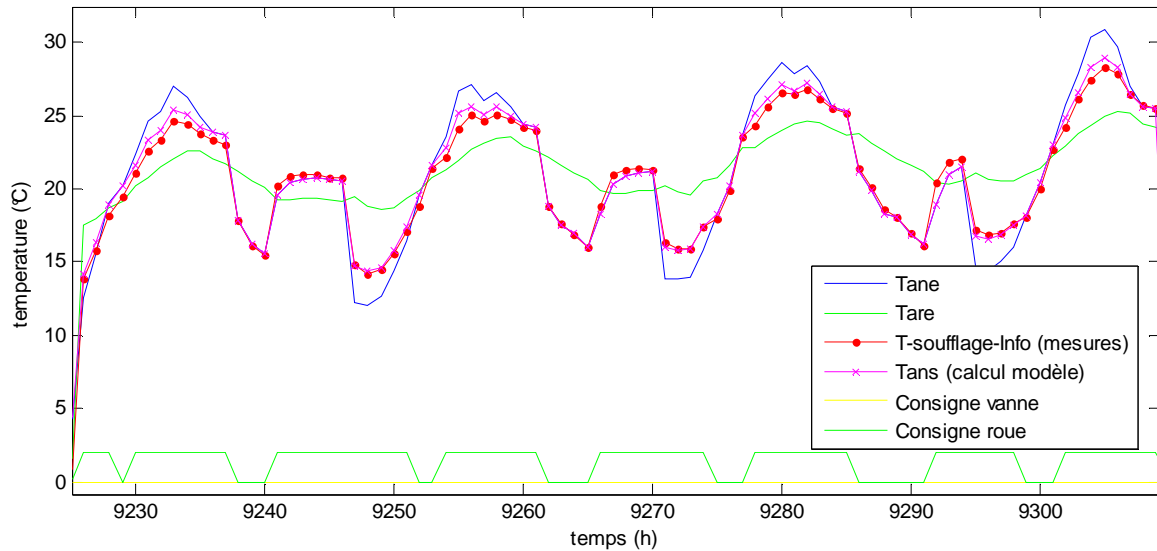


Figure 46. La validation du modèle de l'échangeur rotatif

IV.2.4. La batterie d'eau chaude (BEC)

IV.2.4.1. Caractéristiques du modèle

Concernant la batterie chaude, elle fonctionne en échangeant de la chaleur entre les tubes en aluminium faisant circuler l'eau chaude industrielle, et l'air qui la traverse. Il s'agit d'un échangeur de chaleur à courants croisés avec deux fluides non brassés (Figure 47).

En appliquant la méthode NUT- ε pour une configuration de circulation à courant croisés : le NUT^{nom} peut être déterminé par résolution numérique de l'équation:

$$\varepsilon = 1 - \exp\left(\frac{e^{-NUT * C * n} - 1}{C * n}\right) \text{ avec } n = NUT^{-0,22} \text{ au point nominal.}$$

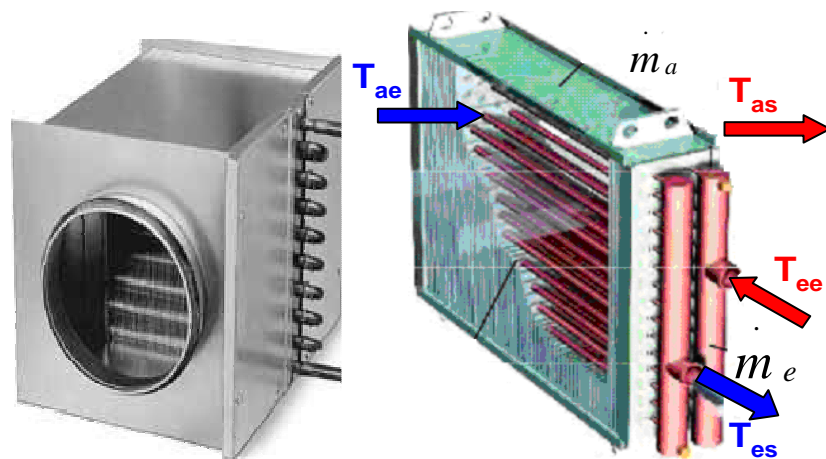


Figure 47. Le modèle de la batterie d'eau chaude

Les coefficients d'échange sont calculés à partir de ceux au point de fonctionnement

en appliquant les relations :

$$\frac{hA_e}{hA_e^{nom}} = \left(\frac{\dot{m}_e}{\dot{m}_e^{nom}}\right)^{0.85} \quad \frac{hA_a}{hA_a^{nom}} = \left(\frac{\dot{m}_a}{\dot{m}_a^{nom}}\right)^{0.72} \quad [\text{WET 1999b}]$$

IV.2.4.2. Implémentation et validation du modèle

Le modèle de la batterie d'eau chaude a été également implémenté dans Matlab et ses entrées définies à partir des mesures de la GTC. Ces entrées sont la température et le débit de l'air (capteurs T_soufflage_Info et D_soufflage_Info) ainsi que la température et le débit de l'eau (capteur Tee_info et le débit est identifié à un point donné et supposé constant) à l'entrée de la batterie.

De la même manière que le modèle de l'échangeur rotatif, la validation du modèle de la batterie d'eau chaude (Figure 48) est effectuée par comparaison entre les résultats de calcul (Tas : courbe rose) et les mesures (T_soufflage_Info : courbe rouge). Étant donné l'absence d'un capteur de température à la sortie de la VMC, nous allons exploiter la mesure de la température d'air au soufflage et le comparer avec les résultats cumulatifs des deux modèles, le premier modèle ayant déjà été validé.

La figure suivante montre que les deux courbes de températures suivent des allures similaires : un réchauffement important de la température de l'air soufflé de manière à se rapprocher des niveaux de température de l'air rejeté. Quelques écarts subsistent, mais qui restent acceptables (un maximum de 2 degrés), lorsqu'on s'éloigne des conditions nominales: température de l'air à l'entrée très basse, température de l'eau très chaude....

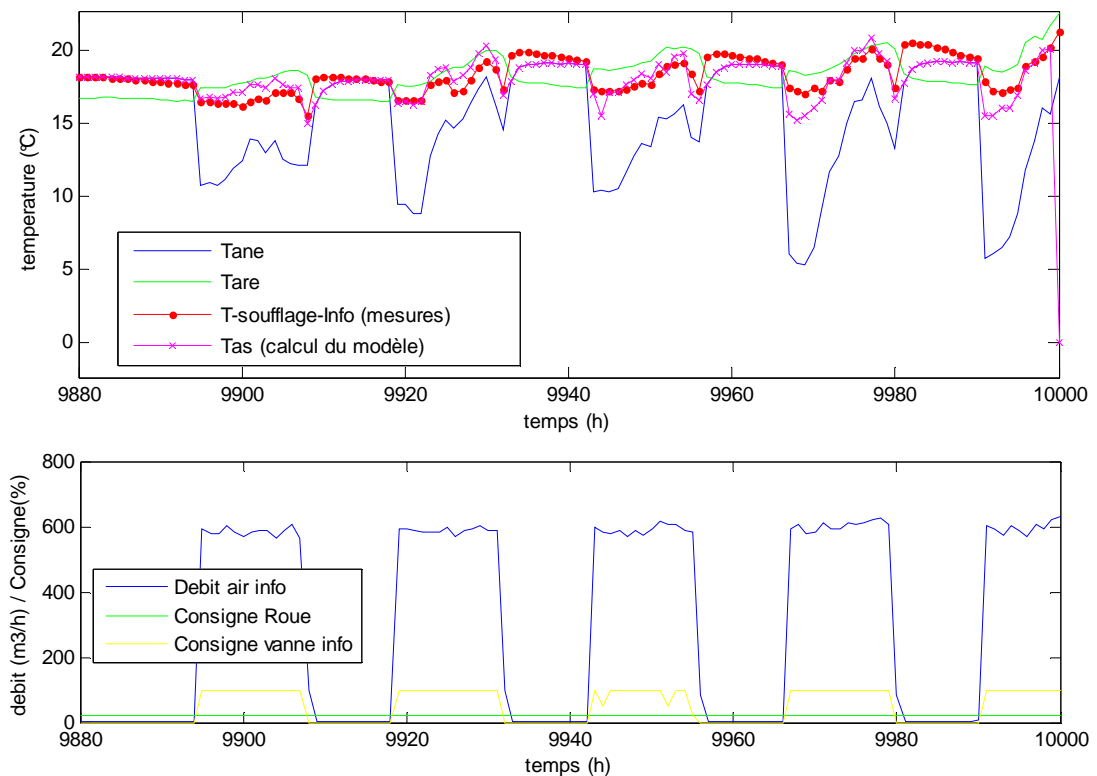


Figure 48. La validation du modèle de la batterie d'eau chaude

L'écart entre la mesure et les résultats de simulation peut s'expliquer par:

- Eloignement du point nominal de fonctionnement. On peut envisager d'adapter ce point de fonctionnement lors des changements importants

- Hypothèses de travail : Le modèle peut être amélioré en tenant compte, par exemple, des variations des propriétés et des états des fluides selon les températures, ou effectuer un calcul plus détaillé des coefficients d'échange. Ces améliorations vont rendre le modèle plus compliqué par rapport aux objectifs d'utilisation.
- Erreur dans les mesures : problème d'étalonnage des températures (température de l'eau), utilisation de plusieurs technologies à la fois, absence de certaines mesures (Le débit d'eau a été identifié pour un point donné et supposé constant car aucune mesure de cette variable n'est disponible actuellement).

D'autres travaux sont envisagés sur la salle PREDIS et qui viendront compléter ce travail, étudier ces hypothèses et améliorer ces résultats.

IV.3. Le circuit aéraulique

Le circuit aéraulique mis en place est composé de plusieurs équipements : les hélices du ventilateurs et le réseau de circulation d'air dont les conduites d'air et les coudes, les bouches d'extraction et de soufflage (Figure 49).



Figure 49. Des composants du circuit aéraulique de PREDIS

La modélisation de ce circuit consiste au calcul des différentes pertes de ces composants qui sont essentiellement des pertes de charge. Elles peuvent être calculées analytiquement au niveau de chaque élément et pour chaque régime d'écoulement (laminaire, turbulent...), ou estimées globalement.

Au niveau du ventilateur (sans moteur), par exemple, on retrouve trois types de pertes [SEMP 2011] (Figure 50) :

- Les pertes mécaniques : dues aux frottements secs et visqueux sur l'arbre du moteur
- Les pertes aérauliques : dues aux pertes de charge
- Les pertes volumétriques : Elles proviennent des circulations d'air au niveau des extrémités des pales provoquées par la différence de pression entre l'intrados (face intérieure des pâles du ventilateurs) et l'extrados (face extérieure exposée au flux d'air).

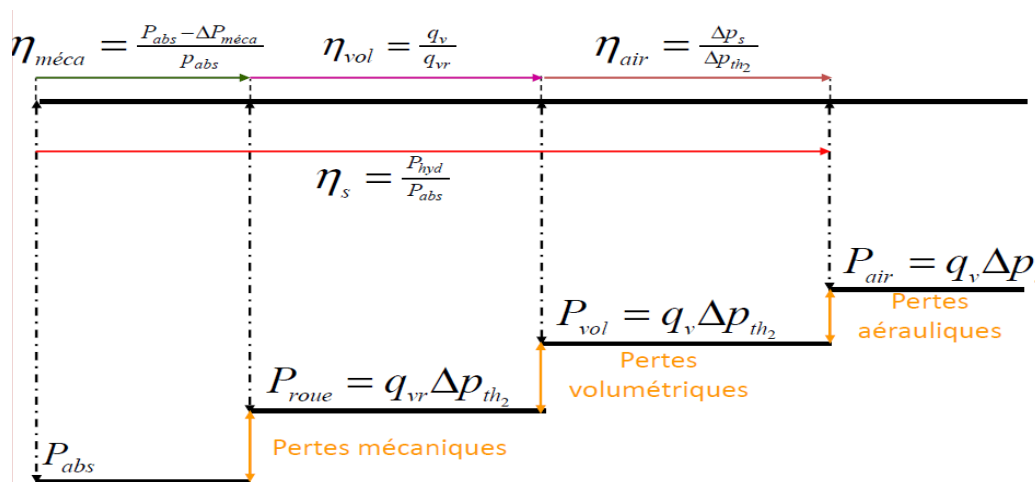


Figure 50. Les différentes pertes du ventilateur

La valeur de ces pertes varie selon le type du ventilateur et son rendement peut être déterminé forfaitairement à partir du tableau suivant (Tableau 4):

Type ventilateur	Rendement
Ventilateur centrifuge à aubes recourbées vers l'arrière (6 à 16 unités)	77 à 80%
Ventilateur centrifuge à aubes recourbées vers l'avant (38 à 42 unités)	57 à 73%
Ventilateur hélicoïde sans diffuseur mais avec redresseur	50 à 88%
Ventilateur hélicoïde avec diffuseur et redresseur	60 à 89%
Ventilateur de paroi	35 à 50%

Tableau 4. Valeur du rendement selon le type de ventilateur

Dans le cas de PREDIS, il s'agit de ventilateur centrifuge à aubes recourbées vers l'arrière (6 unités) donc avec un rendement de l'ordre de 80%.

Nous avons cependant simplifié notre étude en nous basant sur l'identification d'une loi empirique à partir de mesures. La charge aéraulique exerce en effet un couple proportionnel au carré de la vitesse de rotation : $C = k\Omega^2$ (adapté à une commande en V/f à couple variable [DANG 2010]). Ce coefficient de proportionnalité est identifié à partir des courbes de mesures du couple résistant en fonction de la vitesse de rotation (Figure 51) : $k = .1.25 \cdot 10^{-7}$.

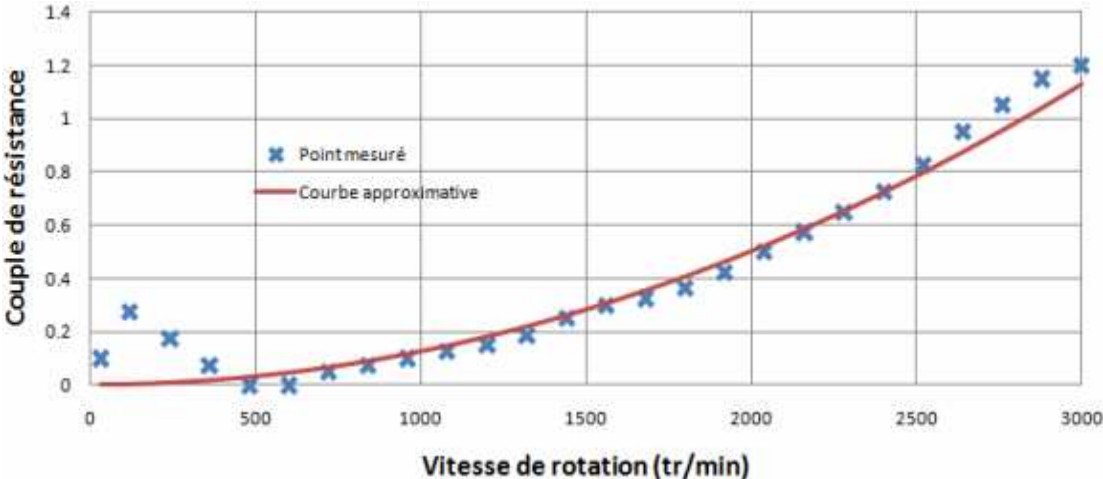


Figure 51. Couple de résistance en fonction de la vitesse de rotation

V. Modélisation des occupants

V.1. Les approches de modélisation du comportement des occupants

La prise en compte de l'occupant devient indispensable dans un contexte de bâtiments sur-isolés et intelligents, équipés de systèmes de gestion énergétique. Sa modélisation en terme d'influence sur l'évolution énergétique dans les bâtiments se focalise sur trois points principaux:

- Son apport thermique dans chaque zone occupée.
- Son usage des appareils installés et l'impact en terme de consommation électrique et production de chaleur.
- Son confort : sa tolérance par rapport aux inconforts et sa sensibilité par rapport à des changement de consignes.

Deux principales méthodes peuvent être employées pour représenter le comportement de l'utilisateur dans chaque type de modèle [PLOI 2012]:

- L'établissement des profils d'occupation et de comportement [BOUR 2005] comme le montre la figure ci dessous (Figure 52) qui présente un profil de variation du pourcentage d'occupation d'un bureau tout au long d'une journée typique (occupation à 50%, par exemple entre, 11h et 1h lors des heures du repas de midi). Ces profils peuvent être établis à partir des traces (remontées des consommations énergétiques ou de la quantité de CO₂ mesurée) ou des calendriers (emploi de temps : pour l'occupation d'une salle de cours) ou également à partir de l'observation du comportement de personnes réelles (observation, questionnaires [Le 2010]). Il s'agit ainsi d'une approche se basant sur l'établissement de scénarios prédéfinis de comportement. Cette approche est utilisée pour décrire les apports des occupants dans COMFIE et également avec le modèle réduit de l'enveloppe.

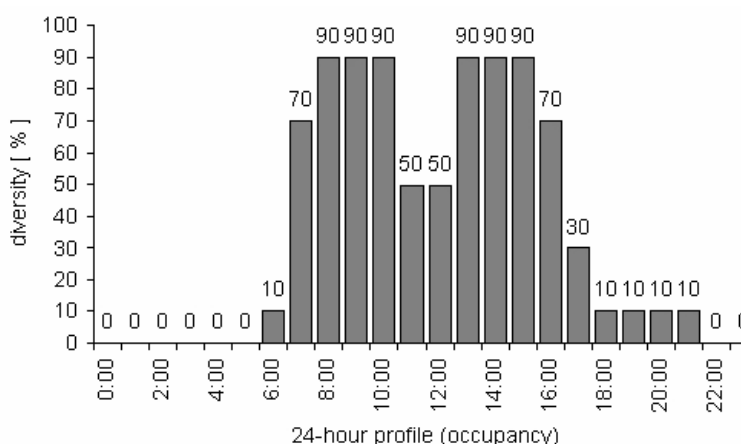


Figure 52. La diversité du profil horaire pour une charge d'occupation typique d'un environnement de type bureau [WINK 1993]

Bien que cette méthode permette de décrire, d'une façon facile à mettre en œuvre, un comportement proche de la réalité des usagers, elle présente des limites au niveau de

l'insensibilité aux variations extérieures et au niveau de l'extension à des nouvelles situations (non étudiées).

- L'établissement de règles de comportement des usagers (cf Brahms) à travers l'utilisation d'agents autonomes, qui « calculent » de manière automatique leur comportement, sans suivre de profils préétablis. Ces agents sont de plus interactifs : ils interagissent avec d'autres agents et avec leur environnement.

I.1. Le modèle multi-agent de l'occupant dans Brahms

V.1.1. Les intérêts de la modélisation multi-agent des occupants

La modélisation multi-agent est largement appliquée aux systèmes sociaux caractérisés par des interactions complexes d'un nombre important d'acteurs complexes [AMB 2011] [MAR 2005] [GILB 2004] [BER 2001]. Elle est utilisée pour divers objectifs : prédiction (ex : évolutions démographique, comportement des consommateurs), compréhensions des erreurs (ex : désastres nucléaire), conception (ex : les issues d'évacuations), divertissement (ex : jeux vidéo)...

Les agents logiciels (Figure 53) représentés par cette approche ont la particularité d'être:

- Autonomes : ils décident des actions qu'ils vont mener (distributed decision making ability [SCH 2003]), à partir des règles établies. Ils peuvent avoir leurs propres pensées, désirs et intentions (BDI) [GEO 1998]...
- Réceptifs et réactifs : conscients de leur environnement et capables d'y réagir.
- Capable de communiquer : peuvent envoyer des messages à d'autres agents.

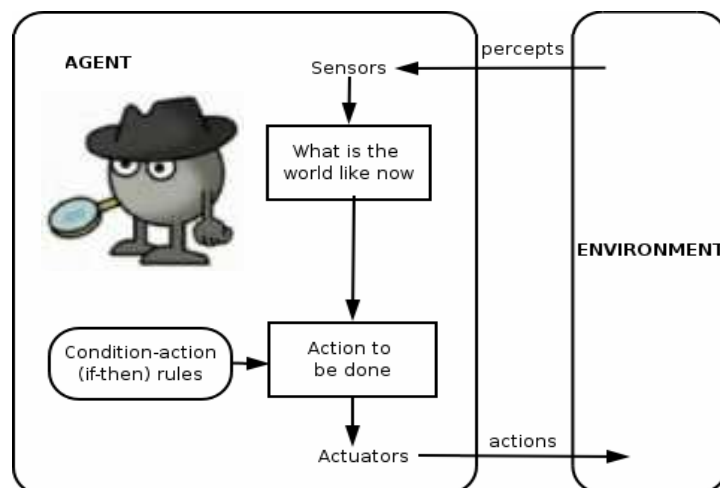


Figure 53. Le concept des agents dans la modélisation multi-agentw

V.1.2. Le modèle de l'occupant dans Brahms

Le comportement humain est modélisé dans Brahms selon le principe des agents et il est basé sur une structure BDI (Belief, Desire, Intention). Un agent (occupant par exemple) est défini, dans le cadre d'un groupe d'agent, par ses activités (exemple : déplacement) qui peuvent être en lien avec d'autres objets, ses connaissances, son emplacement géographique

et des contraintes temporelles qui conditionnent son comportement (Figure 54). Le comportement de l'agent peut être décrit de manière déterministe ou stochastique en spécifiant la probabilité de l'occurrence d'une action donnée.

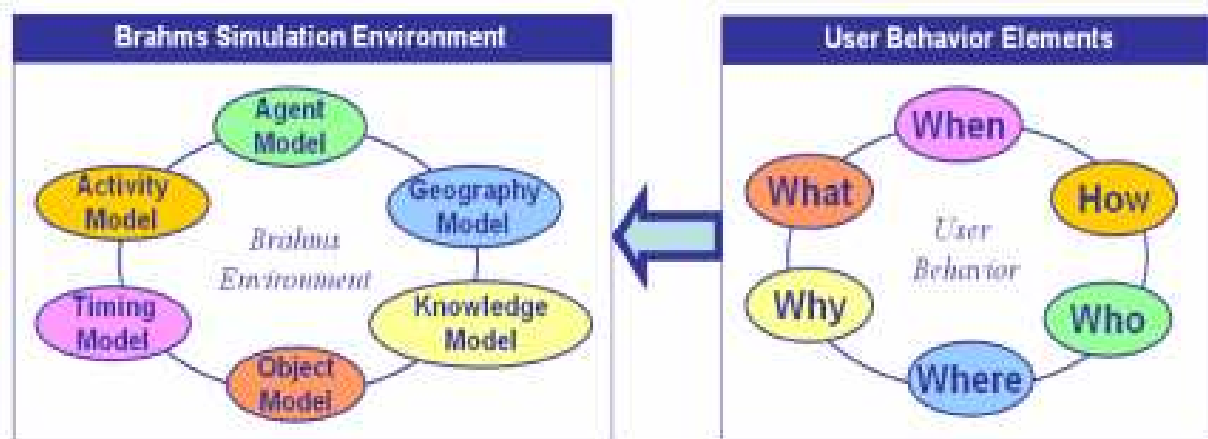


Figure 54. Les éléments de modélisation de l'occupant dans Brahms [KAS 2011].

Dans le cadre du cas test décrit dans la partie 2 du chapitre 4, l'occupant est modélisé par un agent régi par un ensemble de conditions temporelles de déplacement entre les deux zones principales du bâtiment (Zone1 : espace bureaux, Zone2 : salle informatique) et l'extérieur (Ext) (Tableau 5).

Temps	8-10h	10-10h15	10h15-12	12-14	14h-16h	16h-16h15	16h15-18	>18h
Présence	Zone1	Ext	Zone1	Ext	Zone2	Ext	Zone1	Ext
Activité	recherche	pause matin	recherche	pause midi	enseignement	recherche	pause	retour

Tableau 5. Le scénario d'activité décrit dans Brahms

L'occupant est également capable de modifier l'état de l'objet « chauffage » quand il a la sensation de chaleur ou de froid (besoin de chauffer ou de refroidir) selon la température environnante perçue de la zone de présence. On peut ainsi synthétiser l'évolution du comportement de l'occupant étudié comme présenté dans la figure suivante (Figure 55) : selon les facteurs extérieurs (température) et le moment considéré (décidant de la zone occupée), l'agent peut éprouver des besoins (besoin de chauffer) induisant l'exécution d'une activité qui peut induire une autre activité ou un événement sur un autre agent objet (chauffage).

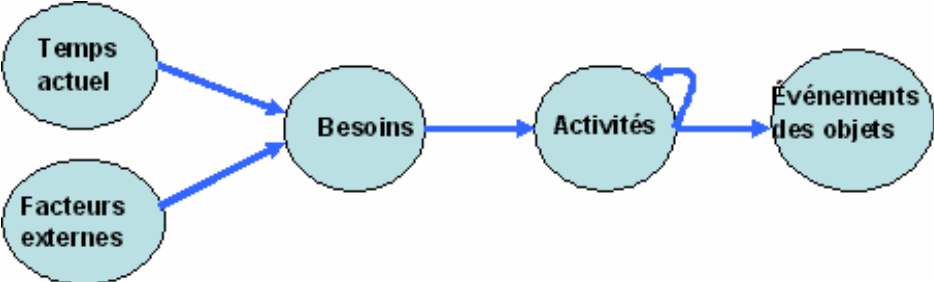


Figure 55. L'évolution du comportement d'un occupant dans Brahms.

VI. Les difficultés rencontrées lors de la modélisation d'un système bâtiment

Une modélisation des différents systèmes de bâtiment est une tâche difficile car divers phénomènes physiques y sont mis en jeu. L'exploitation et l'adaptation des modèles existants doivent être favorisées:

- offrant des modèles assez fiables car développés par des experts du domaine physique traité
- permettant d'économiser l'effort et le temps des nouvelles modélisations et développements.

La modélisation d'un tel système nécessite en même temps des efforts considérables de capitalisation des modèles pour faciliter leurs réutilisations. En contre partie, cette réutilisation est confrontée à plusieurs problèmes lors de leurs adaptations au cas du bâtiment étudié :

- L'adaptation aux objectifs du cas d'étude tel que nous l'avons fait pour le modèle de l'enveloppe issu de Comfie (cf chapitre 4). Ce modèle initialement dédié à la conception de l'enveloppe a été utilisé pour dans une simulation dynamique système (ODE et co-simulation) à des fins de conception des systèmes de gestion de l'énergie.

- L'identification des paramètres du modèle : l'adaptation d'un modèle nécessite généralement l'identification de ses paramètres pour correspondre au système étudié. Cette identification est possible via les données constructeurs ou les données de mesures. Or, les données du constructeur sont généralement difficiles à obtenir et ne sont pas toujours valides pour une installation donnée. Les mesures sont également difficiles à mettre en place car coûteuses et pas toujours très précises (problème d'étalonnage, des perturbations prises en compte...).

VII. Conclusion

Dans ce chapitre, nous avons tout d'abord mis l'accent sur la multiplicité et la diversité des équipements dans un bâtiment, en prenant le bâtiment tertiaire PREDIS comme exemple. Ce phénomène est de plus en plus important dans les nouveaux bâtiments sur-équipés, essentiellement les bâtiments résidentiels (électroménager, multimédia, informatique...). Il devient donc nécessaire d'utiliser souvent plusieurs modèles afin de répondre aux besoins de modélisation de ces différents sous-systèmes. Du fait de la difficulté de modélisation des nombreux et divers équipements, la démarche de réutilisation est souvent pertinente, ce qui est induit néanmoins une remarquable hétérogénéité des modèles utilisés.

Une problématique mise en évidence dans ce chapitre est l'hétérogénéité des modèles exploités dans le secteur du bâtiment. Différentes typologies ont été repérées à travers la modélisation de certains équipements de PREDIS: statiques (échangeur rotatif, batterie d'eau chaude, moteurs) et dynamiques (enveloppe) ; continus (enveloppe : système d'équations n°2, discrets (enveloppe: système d'équations n°3) et événementiels (occupants) ; numérique (Comfie), analytiques (schéma équivalents, échangeurs) et empiriques (aérodynamique, ...). Des paradigmes de modélisation très différents, tels que des modèles à base d'agents et des modèles physiques résolvant des équations aux dérivées partielles.

Dans ce chapitre, les différents modèles composant le système final ont été réalisés, validés et simulés indépendamment. L'aboutissement d'une simulation globale de tout le système nécessite la mise en place de solutions d'interopérabilité afin de pouvoir coupler ces différents modèles tout en tenant compte leurs spécificités et en profitant de leurs interactions. Deux principales approches d'interopérabilité seront étudiées dans le chapitre 3 : une approche boîte blanche se basant sur le langage Modelica et une approche boîte noire se basant sur le concept de composants logiciels et la norme ICAr.

D'un autre côté nous avons souligné plusieurs problèmes d'incertitude relatifs à la modélisation d'un système de bâtiment. Ces incertitudes sont liées à :

- Un manque d'information sur la structure du bâtiment : composition des parois, architecture...
- Une non maîtrise des usages et des conditions météorologiques.
- Des imprécisions et/ou des défaillances des systèmes de mesures (GTC, capteurs).
- Une non connaissance précise du comportement des usagers et des incidences de leurs comportement (cf. typiquement les apports internes liés à leur comportement).

L'objectif de la thèse n'étant pas de se focaliser sur ces incertitudes et de trouver des solutions performantes pour améliorer la modélisation de ces systèmes, ces aspects seront étudiés avec plus de détails dans d'autres projets de recherche sur le bâtiment PREDIS (thèse en cours de Mathilde Grandjacques : « Méthodes et outils pour la propagation d'incertitudes des simulations en énergétique des bâtiments ») ou sur d'autres bâtiments (projet ANR

fiabilité). Par contre, c'est grâce aux solutions d'interopérabilités mises en place dans ce travail de thèse (chapitre 3 et 4) que ces problématiques pourront être mieux étudiées et achevées au niveau des systèmes.

Chapitre 3 :

Les approches de modélisation système: application aux systèmes de PREDIS

I. Introduction

L'Ingénierie Système (IS) est une approche méthodologique interdisciplinaire de formalisation de l'ensemble des activités autour du cycle de vie des systèmes complexes : conception, vérification, amélioration [REN 2011]. Ce domaine de l'informatique se base sur des modèles (Ingénierie Dirigée par les Modèles : IDM ou Model-Driven Engineering : MDE) pour représenter les connaissances des différentes composantes du système. Il s'agit de développer des outils logiciels, des concepts informatiques et des langages de description (paradigmes, plateformes, méthodes, standards...) pour créer, transformer et utiliser facilement des modèles.

L'application de ces formalismes et techniques élaborés dans l'ingénierie dirigée par les modèles, l'ingénierie système et l'ingénierie logicielle (*software engineering*), tend à se généraliser dans les domaines physiques multidisciplinaires comme l'aéronautique ou l'automobile. Son application au domaine du bâtiment semble pertinente, pourtant elle n'a pas encore réellement percé en raison peut être d'un manque de standardisation des moyens d'interopérabilité.

En effet, ces méthodes sont employées entre autre pour répondre aux problématiques d'hétérogénéités des modèles, dépasser les limites des outils et apporter plus de facilité d'interopérabilité. Deux principales techniques seront exploitées dans ce travail en application au secteur du bâtiment (plus spécifiquement au cas de PREDIS) pour satisfaire aux besoins exprimés dans le chapitre 1 et illustrés dans le chapitre 2.

La première partie que nous allons présenter repose sur un partage de l'information des métiers relatifs à chaque modèle à l'aide de langages de modélisation multi-métiers et généralistes. Dans le secteur du bâtiment, le standard Modelica connaît un essor important dans la modélisation des équipements et du système. Il sera ainsi adopté pour la modélisation du système de bâtiment PREDIS en vue de mettre en valeur les avantages de l'application d'une telle approche dans ce secteur. Les limites de cette approche seront également illustrées pour conclure sur la nécessité de l'utilisation d'une approche complémentaire.

Une deuxième approche va s'appuyer sur le paradigme de composants logiciels, une architecture indépendante des applications et des outils de simulations. Plusieurs standards de composants ont été définis ces dernières années dans plusieurs secteurs de l'ingénierie. Nous proposons, dans ce manuscrit, l'adoption du standard ICAr, développé au sein du laboratoire pour des applications en génie électrique [DEL 2003], pour la modélisation des systèmes du bâtiment. Cette technique voit son intérêt grâce à son architecture multi-facettes et à deux concepts de base complémentaires « plug-in » et « plug-out ». Ces concepts vont permettre l'amélioration de la portabilité et la réutilisabilité des modèles en assurant une génération et une utilisation automatique des composants logiciels depuis et vers différents outils, dont les environnements du langage Modelica.

Ces deux approches vont participer à la capitalisation des modèles dans des bibliothèques distribuables [WUR 2008] et apporter plus de facilité d'échange de modèles entre collaborateurs grâce à un format unique de description.

II. L'approche « boîte blanche » de modélisation système se basant sur le langage Modelica

II.1.L'approche « boîte blanche » de modélisation système

La modélisation système se basant sur l'approche boîte blanche consiste en la description des différents modèles constituant le système dans un langage unifié de modélisation selon un ensemble de règles bien définies. Il s'agit de représenter graphiquement ou textuellement les équations mathématiques du modèle qui restent accessibles aux utilisateurs, d'où l'appellation « boîte blanche ».

Plusieurs langages de modélisation ont été définis pour faciliter la description des systèmes. Plusieurs études comparatives entre les différents langages [GUI 2007] [THEI 2009] ont été réalisées afin de repérer leurs principales différences et particularités. Elles se basent sur la comparaison de leurs caractéristiques (normalisation, origine, modularité...), les approches de modélisations employées (discret/continu, explicite/implicite, ODE/DAE...) ainsi que certaines syntaxes (ports, gestion des discontinués...). Au sein de notre équipe de recherche, on s'est essentiellement intéressé à deux langages de description dans le cadre de cette approche de modélisation: VHDL-AMS et Modelica.

VHDL-AMS [IEEE 1999] est l'extension du VHDL, initialement développé pour la modélisation des circuits d'électronique numérique, permettant de décrire leurs comportements numériques et analogiques. Ce langage est étudié avec plus de détails dans le cadre de la thèse d'Abir Rezgui [REZ 2012] qui s'est déroulée en parallèle de la mienne, en application aux systèmes mécatroniques. Ce langage présente certains avantages comme un cadre normatif [IEEE 1999], la possibilité de modélisation idéale des composants de commutation (diode, interrupteur...) à travers la notion de « across »/« through » [FUR 2009], indispensable dans le domaine électronique. Cependant, son utilisation reste confrontée à la limitation du nombre de bibliothèques de modèles existantes et à des limitations des outils de simulation, où la norme reste partiellement implémentée [REZ 2010]. En outre, la notion d'héritage existante dans Modelica n'y est pas définie, elle constitue pourtant un atout important pour la réutilisation des modèles.

Le langage Modelica est dédié à des applications génériques et à des systèmes physiques complexes et multidisciplinaires, ne nécessitant pas un haut niveau de précision de description. De plus, il présente l'avantage d'une large communauté (Modelica association [WEB_MOD]) très active. Cette communauté a permis la multiplication et la capitalisation des bibliothèques de modèles ainsi que des environnements de simulation (cf chapitre 1) [POP 2008]. En outre, il s'agit d'un langage orienté objet offrant la possibilité d'exploiter les concepts d'héritage et de profiter ainsi de facilité de réutilisation.

Ce langage sera adopté dans la suite pour la modélisation du système de bâtiment PREDIS. En effet, Modelica semble à présent être adopté dans la communauté du bâtiment et un consensus semble se former autour de ce langage [WET 2012]. Par contre, le débat entre

VHDL-AMS et Modelica se pose encore dans d'autres domaines comme l'aéronautique et l'automobile [EJJ 2010] [HAM 2012].

II.2. La modélisation dans Modelica

II.2.1. Les principales caractéristiques du langage

II.2.1.1. Un langage multi-physique

Modelica [FRI 2004] est un langage multidisciplinaire pouvant couvrir plusieurs domaines physiques. Cette caractéristique s'avère indispensable pour la modélisation des systèmes complexes composés de plusieurs sous systèmes dérivant de différents domaines comme c'est le cas dans l'aéronautique, l'automobile et le bâtiment... La bibliothèque standard de Modelica vient ainsi couvrir plusieurs domaines physiques : électrique (analogique et numérique), magnétique, mécanique, fluide, thermique, contrôle ...

Chaque composant d'un domaine physique est défini par des ports énergétiques spécifiques, en plus des signaux. Un signal est un message codé de communication pouvant être logique ou analogique sans nécessairement avoir une signification physique. L'information y circule dans un sens bien défini et connu par avance; de l'expéditeur vers le destinataire. Alors qu'un port énergétique représente une connexion caractérisée par deux variables physiques d'effort et de flux, selon le principe des « bond-graph » [KAR 2000] [ROB] ou des variables effort/flux modifiées comme utilisées dans le domaine mécanique (Tableau 3). La variable de flux peut changer de signe et de sens de circulation [FRIT 2006b].

Domaine	Variable d'effort	Variable de flux
Electrique	Tension	Courant
Mécanique, translation	Position	Force
Mécanique, rotation	Angle	Couple
hydraulique	Pression	Flux volumique
Thermique	Température	Flux d'entropie
Magnétique	Force magnétomotrice	dérivée du flux magnétique

Tableau 6. Les variables efforts flux de quelques domaines physiques dans Modelica

II.2.1.2. Un langage basé sur des équations et orienté objet

Modelica est à la fois un langage orienté objet et basé sur des équations : EOO (Equation Oriented Object language) [BRO 2006].

En effet, un modèle Modelica est décrit par un ensemble d'équations (ou algorithmes) définissant son comportement. La description de ces équations est sous la forme de DAE implicite [BRO 2007] pouvant être transformés en ODE lors de la compilation (cf paragraphe suivant). Les modèles continus, discrets et hybrides sont supportés par ce langage.

$$f(\dot{X}(t), X(t), U(t), p, t) = 0$$

D'un autre côté, il s'agit d'un langage orienté objet. La programmation orientée objet est un paradigme de programmation informatique qui a vu son apparition au début des années 60 [MCC 1960]. Il se base sur le concept d'un objet, entité physique, et la définition de son contenu et son interaction avec d'autres objets. Ce paradigme de modélisation présente deux principaux intérêts :

- Un prototypage plus rapide des systèmes.
- Une réutilisation plus aisée des composants du système.

Ce principe de modélisation adopté par Modelica offre un prototypage rapide des systèmes complexes [CEL 1996]. En effet, un système peut être décomposé en multiples sous systèmes qui peuvent être décrits et traités de manière séparée puis ensuite rassemblés pour construire le système global. Cette analyse séquentielle permet une structuration des données du modèle facilitant sa compréhension et son exploitation par les utilisateurs.

Ce langage assure également une réutilisation aisée des modèles grâce aux concepts d'héritage et de polymorphisme grâce au mot clé « extends ». Des nouveaux modèles peuvent être ainsi créés par extension des modèles existants [BRO 2007]. Le module « DryEffectivenessNTU » utilisé pour la modélisation de la batterie d'eau chaude, par exemple, a été hérité du modèle « PartialEffectiveness » (Figure 56).

```
model DryEffectivenessNTU
  "Heat exchanger with effectiveness - NTU relation and not moisture condensation"
  extends Buildings.Fluid.HeatExchangers.BaseClasses.PartialEffectiveness(sensibleOnly1 = true,sensibleOnly2 = true);
```

Figure 56. Exemple d'héritage dans Modelica

II.2.1.3. Une représentation causale et/ou acausale

Le langage Modelica permet de réaliser à la fois une description causale (block) et acausale des modèles (« boule » [ALL 2003] [DEL 2012b]) [ELM 1997]. Un modèle causal est caractérisé par des connexions de type signaux unidirectionnels alors que le modèle acausal possède des ports énergétiques.

La modélisation acausale présente des avantages par rapport à la description causale car elle offre une description :

- Plus naturelle [FRIT 2006b]: cette description est proche de la réalité car les connexions y représentent un transit des variables énergétiques (efforts flux) entre les modèles. Le positionnement des modèles correspond mieux aux emplacements physiques des équipements correspondants. La séparation et la délocalisation de certains modèles pour garantir la causalité n'est plus nécessaire (Figure 60 et Figure 62), ce qui peut rendre la modélisation plus lisible et compréhensible par l'utilisateur.

- plus flexible: l'introduction d'un nouveau modèle ne nécessite par le re-ordonnement manuel des équations du système pour établir la nouvelle causalité. Il suffit ainsi de le connecter aux autres composants, la causalité sera gérée automatiquement.

- plus automatique: la causalité de chaque modèle n'est plus traitée au préalable par transformation sous forme d'une représentation d'état. Il suffit de décrire le modèle à travers des équations différentielles algébriques (DAE) et la causalité du système sera traitée dans sa globalité par le compilateur. Un compilateur « Modelica » assure l'ordonnancement de toutes les équations du système et la transformation du système acausal en système causal par le traitement des boucles algébriques et différentielles.

La plupart des compilateurs Modelica permettent le traitement d'un modèle pour en générer un code informatique en langage C équivalent, en passant par les principales étapes suivantes [FRIT 2009] (Figure 57) :

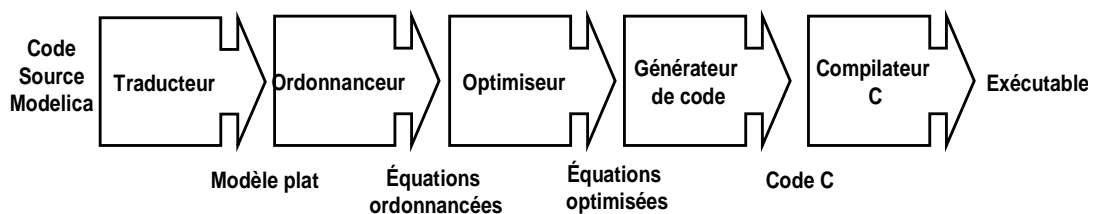


Figure 57. Etapes de compilation d'un code Modelica

Les différentes phases de la compilation d'un modèle se détaillent de la façon suivante :

-Traduction du modèle : il s'agit d'analyser le modèle, instancier tous les objets référencés par le modèle, traduire les connections par les équations équivalentes et traiter les différents héritages entre les modèles pour aboutir à un unique modèle plat.

-Ordonnancement des équations : au cours de cette étape, toutes les équations seront ordonnées selon les dépendances des flux d'informations. Dans le cas des équations DAE, une manipulation de transformation en matrice triangulaire est effectuée. Cette étape peut être implémentée dans certains compilateurs pour assurer la transformation des DAE implicites en ODE explicites tel est le cas des compilateurs OpenModelica [FRIT 2001] et Dymola [ZIM 2009].

-L'optimisation des équations: c'est un module permettant de simplifier les équations et réduire leurs nombre grâce à des méthodes mathématiques.

-La génération du code « C » et sa compilation: Il s'agit de la génération du code « C » correspondant au modèle Modelica après les opérations de simplifications précédentes. Ce code C est ensuite compilé en code binaire pouvant être appelé par le solveur de l'outil.

II.2.2. La structure des modèles en Modelica

II.2.2.1. La structuration d'un modèle

Un modèle Modelica est composé, comme le montre la figure ci-dessous (Figure 58), de deux parties principales :

- L'interface du modèle: cette partie permet de définir le modèle et sa connexion avec l'extérieur ; type de la classe (model, block...) et son nom, les ports, les héritages (bibliothèque), instances des composants, les paramètres, les variables...

- Le corps du modèle : cette partie permet de décrire le comportement du modèle à travers des équations différentielles et/ou des algorithmes

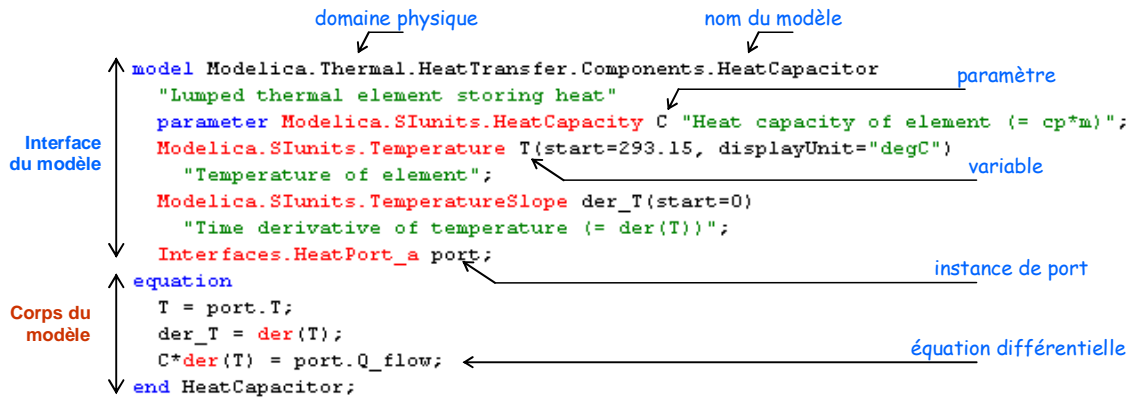


Figure 58. La structure d'un modèle de capacité thermique dans Modelica

II.2.2.2. Les connexions entre les modèles

Chaque modèle est défini par un ou plusieurs ports, pouvant avoir des natures différentes, qu'ils exposent pour une connexion externe avec d'autres modèles. Le couplage entre deux modèles est effectué par une connexion graphique ou textuelle entre des ports qui doivent être de même nature (même domaine et par conséquent même variables efforts/flux). Textuellement, les spécifications de Modelica [SPEC 2010] définissent un mot clé « connect » qui permet de connecter deux ports de deux modèles. Ce concept est traduit par le compilateur par les deux équations de Kirchhoff correspondantes (Figure 59).

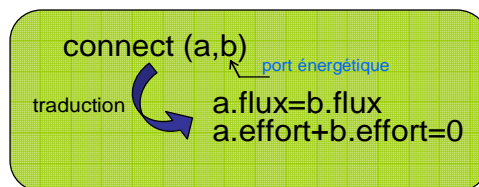


Figure 59. Les connexions dans Modelica

II.3.L'approche d'interopérabilité se basant sur le langage Modelica en application à la modélisation du système de bâtiment PREDIS

Cette solution d'interopérabilité repose sur le développement de tous les sous-systèmes constituant le système global, qui dérivent de différents domaines, dans le même langage Modelica [KOS 2006].

Dans la suite, nous allons procéder à la projection des différents modèles de la salle PREDIS MHI présentés dans le chapitre 2, en langage Modelica dans l'environnement Dymola. Il s'agit principalement d'exploiter des modèles existants à partir de la bibliothèque standard et également de la bibliothèque « buildings » [WET 2009], et de les adapter aux besoins spécifiques pour la construction de l'ensemble des modèles de PREDIS.

Pour éviter le re-développement de tous les modèles, qui s'avère une tâche fastidieuse, la réutilisation des bibliothèques existantes sera privilégiée.

II.3.1. Description des modèles de PREDIS dans Modelica

II.3.1.1. Modélisation de la VMC dans Modelica

II.3.1.1.1. La partie électrique (MAS + variateurs)

Dans un premier temps, un modèle causal (Figure 60) a été développé à partir des modèles causaux réalisés dans Mathcad (cf chapitre 2). En se basant sur les interfaces du domaine « blocks » de la bibliothèque standard, les modèles ont été entièrement re-développés dans Modelica et décrits à travers les équations et les algorithmes correspondants.

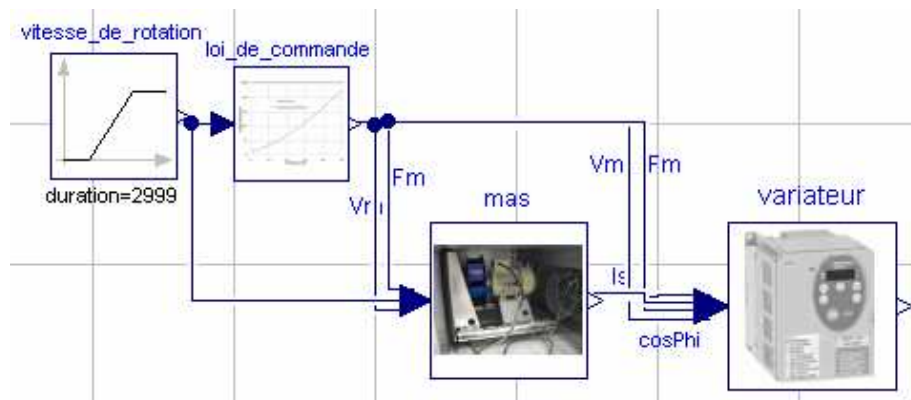


Figure 60. Modélisation causale de la partie électrique de la VMC dans Modelica

Chaque élément a été vérifié séparément pour comparer les résultats obtenus dans Modelica à ceux précédemment obtenus dans Mathcad. Pour la machine asynchrone, par exemple, on observe les courbes de variation du rendement et du couple (Figure 61).

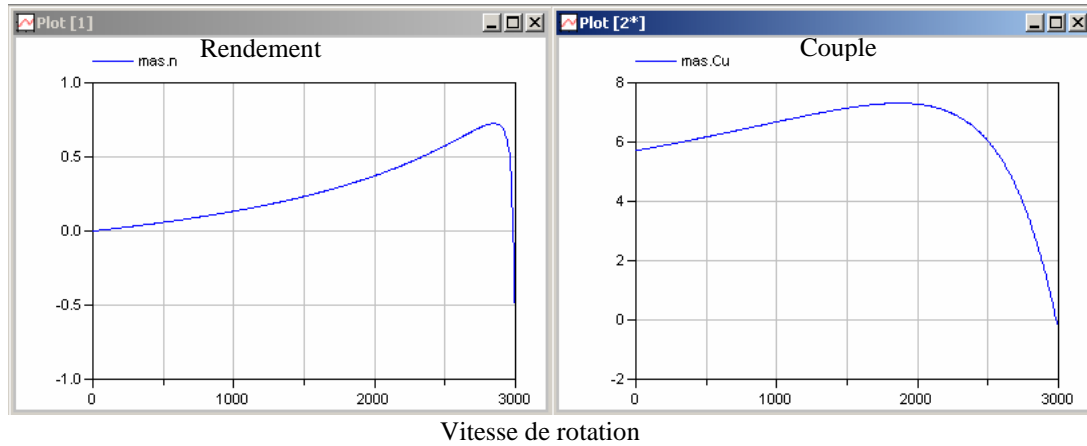


Figure 61. Résultats de simulation du modèle causal de la MAS de la VMC dans Modelica

Un modèle acausal a été également établi afin de garantir plus de flexibilité au modèle et montrer l'intérêt de cette approche de modélisation (Figure 62). Cette représentation est plus naturelle vue qu'elle correspond plus à une réalité physique. Il n'est plus nécessaire de séparer la loi de commande du block de variateur de vitesse pour éviter la boucle algébrique (le modèle du variateur a besoin du facteur de puissance qui doit être calculé par la machine, qui à son tour a besoin de la fréquence de modulation ainsi que du courant et de la tension). C'est le compilateur Modelica qui se chargera de la gestion de la causalité du système entier et du traitement de ses boucles algébriques. Les résultats obtenus sont similaires à ceux de la Figure 61.

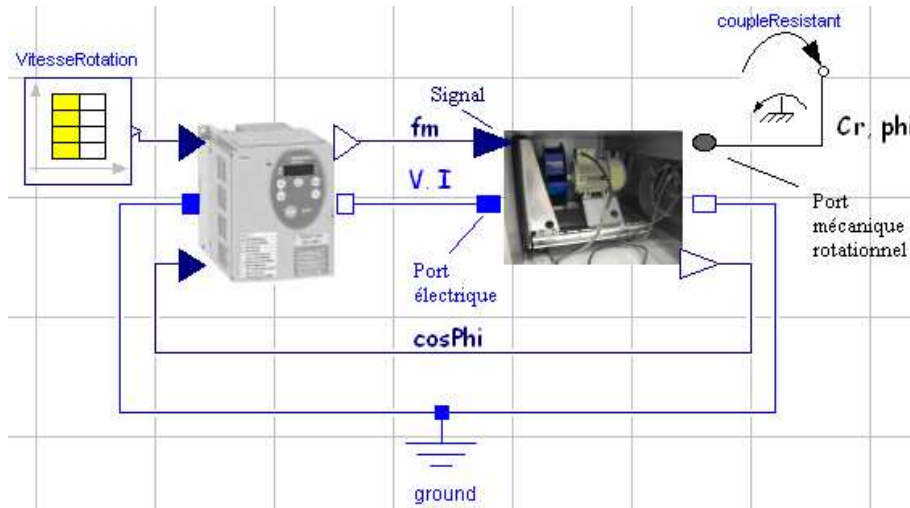


Figure 62. Modélisation acausale de la partie électrique de la VMC dans Modelica

II.3.1.1.2. Les échangeurs de chaleurs

Afin de modéliser les échangeurs de chaleur de la salle PREDIS dans Modelica, le module «DryEffectivenessNTU» de la bibliothèque «Buildings» (Buildings.Fluid.HeatExchangers) a été utilisé. Il se base sur le même principe de modélisation décrit dans le chapitre 2 (méthode NUT- efficacité) avec certaines différences nécessitant quelques adaptations des paramètres pour mieux coller aux résultats des deux modèles.

Le modèle de la batterie d'eau chaude a été directement construit à partir de ce module après adaptation des différents paramètres (natures des fluides, configurations, conditions nominales).

Bien que ce module couvre plusieurs types et configurations d'échangeurs, il n'est adapté qu'à des échangeurs statiques. Il ne peut pas prendre en compte la rotation d'un échangeur. Ainsi, un nouveau modèle a été développé pour décrire l'échangeur rotatif, à partir de ce module, en rajoutant les termes et les équations représentant la rotation de la roue (Figure 63). Ce modèle peut être rajouté dans la bibliothèque «buildings».

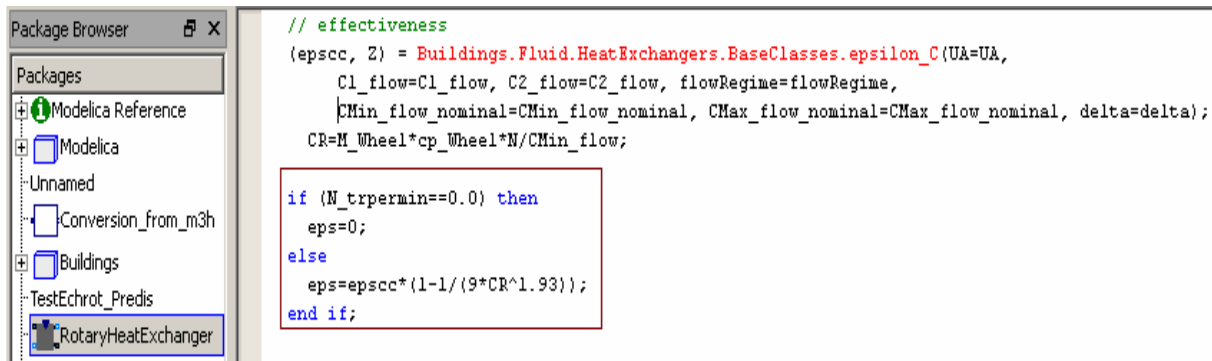


Figure 63. L'adaptation du modèle de l'échangeur en Modelica pour supporter la rotation

Chaque modèle est ensuite vérifié par comparaison aux résultats obtenus dans Matlab au chapitre 2, pour la même période de test et les mêmes données de la GTC. Pour les modèles de l'échangeur rotatif par exemple, les résultats obtenus dans Dymola et Matlab correspondent sur la majorité de la période de mesure (sans grande variation) (Figure 64).

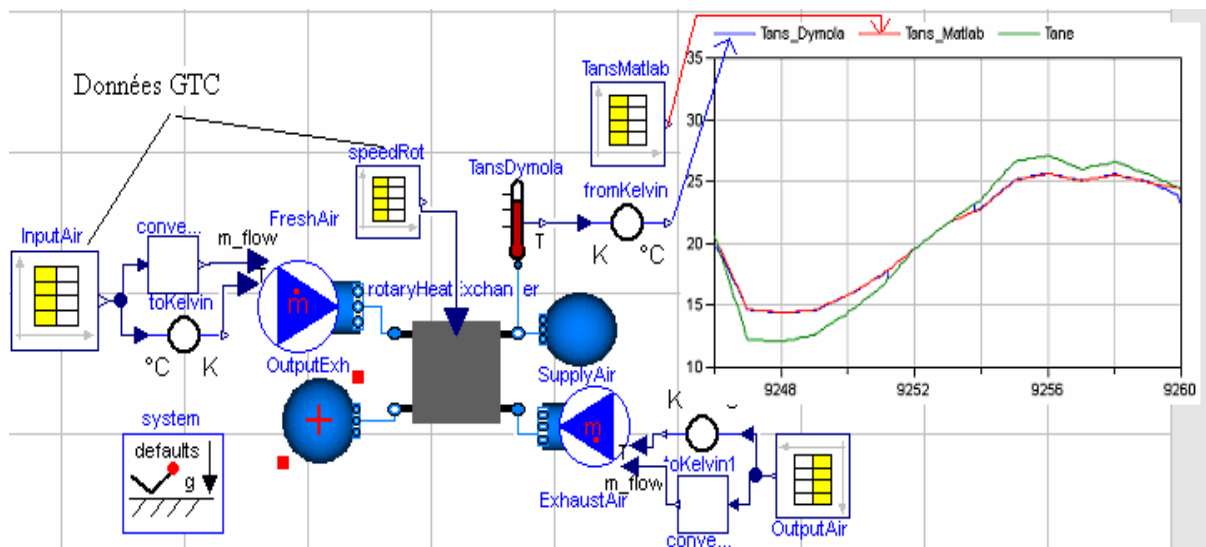


Figure 64. Projection du modèle de l'échangeur rotatif dans Modelica avec mise en oeuvre dans Dymola

De la même manière, le modèle de la batterie d'eau chaude a été vérifié ainsi que le modèle des pertes du circuit aéraulique. Ces derniers permettant de déterminer à partir d'une consigne globale de débit dans tout l'espace, le débit soufflé dans chaque zone.

Les différents modèles ont été ensuite couplés, comme présenté dans la figure ci-dessus (Figure 65), pour représenter la circulation de l'air et l'échange de chaleur dans la VMC. Les résultats du système globale dans Dymola ont été également comparés à ceux obtenus dans Matlab (Figure 66).

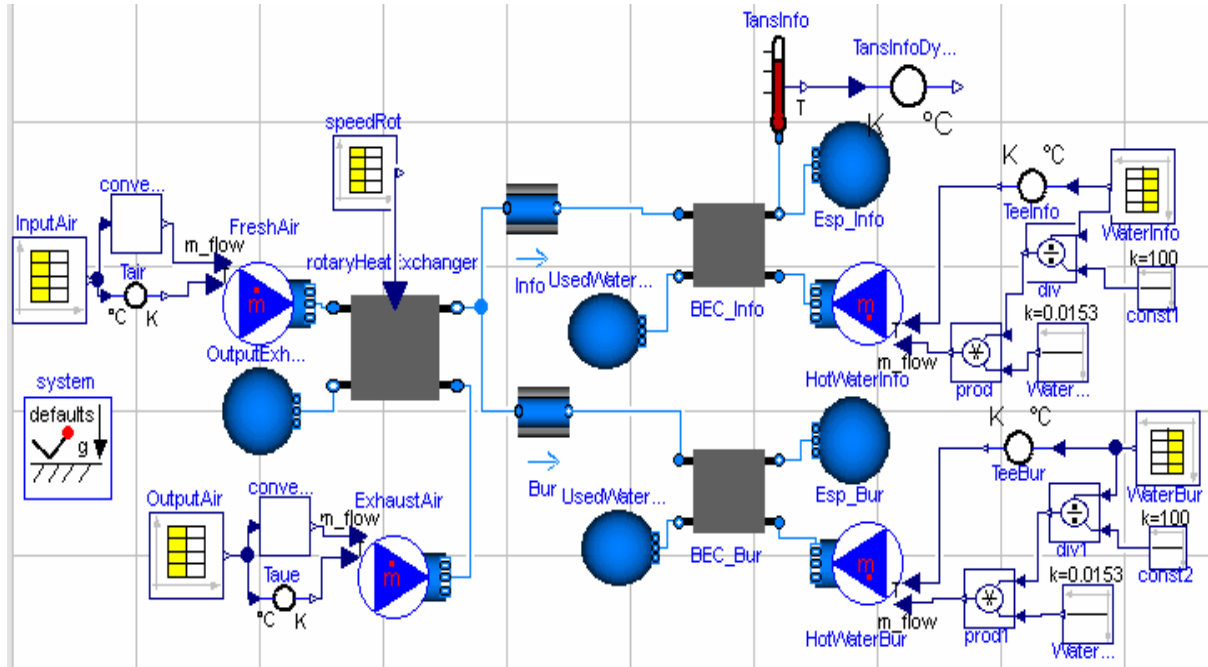


Figure 65. Le modèle de la VMC de PREDIS dans Modelica

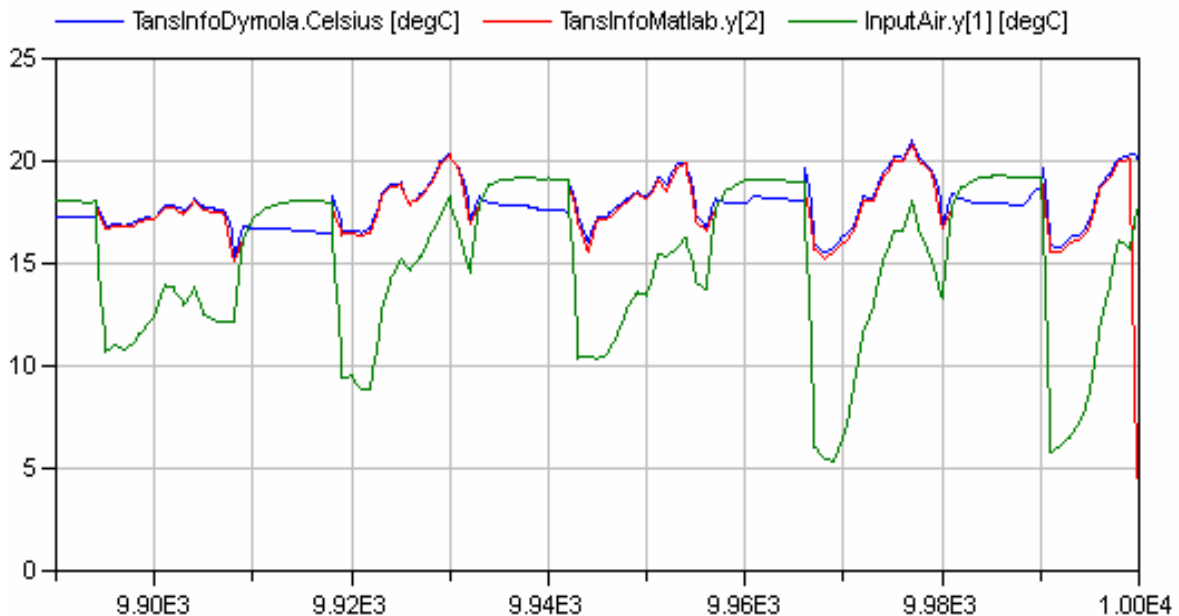


Figure 66. Résultats de la projection des modèles de la VMC dans Modelica

II.3.1.2. Modélisation de l'enveloppe et des usages dans Modelica

II.3.1.2.1. Le modèle réduit et les profils d'usage

Le modèle réduit de l'enveloppe a été construit par connexion des différents composants thermiques issus de la bibliothèque standard (capacités, résistances, sources de flux...). Les paramètres préalablement identifiés (cf chapitre 2) ont été affectés à chaque

composant de manière à obtenir les mêmes résultats que dans Matlab Simulink (Figure 67). Les occupants ainsi que les équipements tels que la VMC, sont définis par des profils prédéterminés de puissances comme décrit dans le chapitre 2.

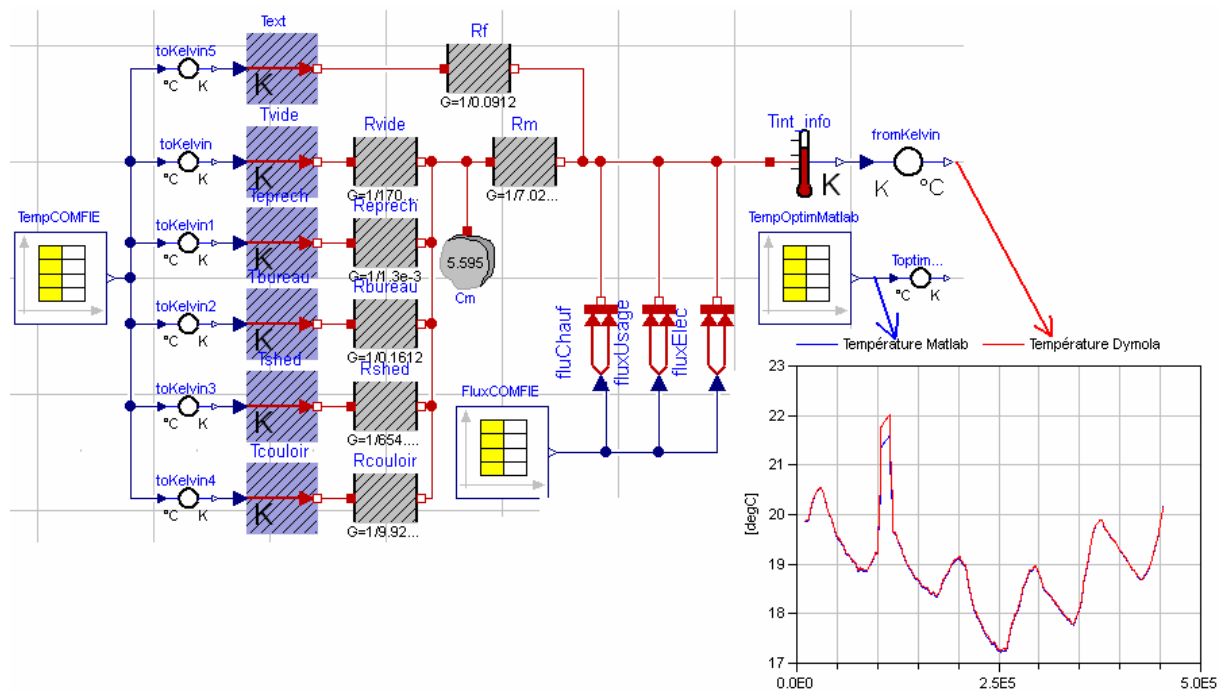


Figure 67. Projection du modèle de l'enveloppe et des usages dans Modelica

II.3.1.2.2. Les améliorations envisageables du modèle d'usage

Plusieurs améliorations peuvent être apportées à ce modèle pour tenir compte de manière plus précise des usages du bâtiment. On peut par exemple intégrer le modèle de la VMC au lieu d'utiliser de simples profils. Ce modèle intervient essentiellement au niveau des sources de flux thermique de l'enveloppe. La quantité de chaleur due à la ventilation est injectée dans le modèle de l'enveloppe et vient influencer son comportement thermique en fonction de la température extérieure en particulier (Figure 68). La VMC influence également le bilan énergétique du bâtiment à cause des puissances électriques consommées par les systèmes électriques.

Grâce aux facilités de modélisation offertes par le langage Modelica, on peut également introduire des modèles des ordinateurs portables installés dans les deux salles de PREDIS. Un modèle de ces batteries a été réalisé dans le cadre de la Thèse de Dang [DANG 2012]. L'intégration de nouveaux systèmes dans PREDIS, comme des panneaux solaires ou une pompe à chaleur (PAC), peut être facilement pris en compte dans le système global afin d'étudier en simulation l'impact de différentes configurations envisagées.

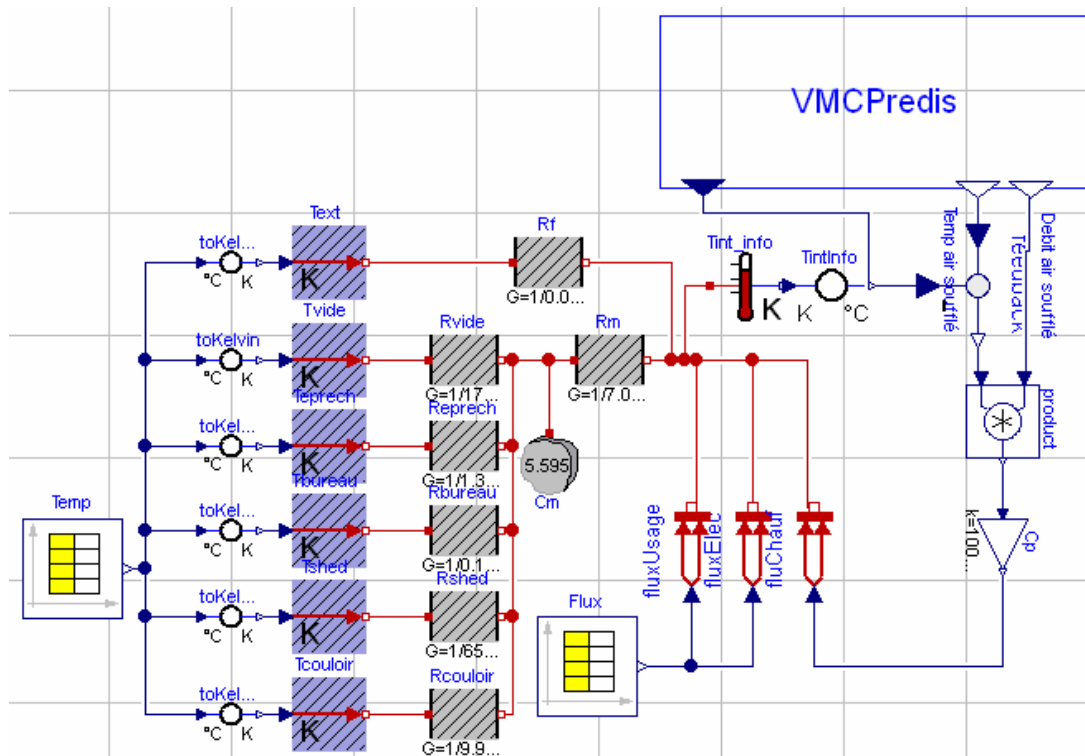


Figure 68. Le couplage entre la VMC et l'enveloppe du bâtiment PREDIS dans Modelica

Il ne nous a pas été possible d'intégrer à ce modèle Modelica un modèle plus détaillé des occupants, se basant sur une approche multi-agent, développé dans Brahms. Car, d'un côté Brahms ne permet pas l'extraction de ses modèles pour un traitement externe et d'un autre côté il n'est pas possible de transposer directement ce formalisme de modélisation dans le formalisme de Modelica.

Une modélisation plus fine de l'enveloppe se basant sur le principe des volumes finis adopté dans COMFIE, s'avère aussi délicate car Dymola ne permet pas une résolution performante des PDEs.

Pour dépasser les limites de cette approche purement « boîte blanche », et permettre une intégration aisée de modèles reposant sur d'autres formalismes que Modelica, nous proposons par la suite un couplage avec l'approche « boîte noire ».

De manière duale, nous souhaiterions exploiter les modèles Modelica pour d'autres applications que la simulation dynamique et donc exploiter ces modèles dans d'autres outils. Il peut s'agir de Matlab pour tester des stratégies avancées de contrôle commande, comme la gestion de la charge et la décharge de la batterie électrique ; ou encore d'un outil dédié à l'optimisation comme CADES [DEL 2004] permettant de déterminer la commande optimale des moteurs de soufflage [DANG 2011].

II.4. Les limites de cette approche d'interopérabilité

L'utilisation du langage Modelica est une solution efficace d'interopérabilité, cependant, cette approche présente certaines limites sur plusieurs niveaux : théorique, technique et pratique.

- Limite théorique :

Il est théoriquement impossible, selon le théorème d'incomplétude de Gödel [GOD 1992], de construire un langage formel, consistant et complet pour tous les types de simulations physiques. Ce langage n'est par exemple pas compatible avec une modélisation multi-agents (e.g. Brahms).

- Limites techniques :

En outre, divers problèmes techniques reliés aux simulateurs de ce langage ont été rencontrés. Il s'agit par exemple de difficultés de résolution des équations aux dérivées partielles (PDE) [MAZ 2009]. En effet une extension du langage pour le support des PDE a été formulée [SAL 2000] [SAL 2002] mais son implémentation et sa traduction restent limitées [LI 2009]. Des problèmes de convergence ont été également repérés dans certains cas d'étude comme pour la simulation des bâtiments multi-zones dans Dymola [WET 2006] et la détection des bugs reste délicate. En effet, dans ces langages où la description d'équations complexes se fait de manière aisée, plusieurs missions se trouvent confiées aux noyaux de calcul (gestion de la causalité, des boucles algébrique, transformation des PDE et des DAE en ODE...). L'inconvénient de cette sous-traitance est que les difficultés de résolution sont ainsi éclipsées au programmeur qui devient incapable d'intervenir pour la correction ou même le repérage des problèmes [THE 2009].

- Limites pratiques :

Pour de multiples raisons pratiques, la solution basée sur l'approche boîte blanche ne peut pas, et ne doit pas être la solution unique d'interopérabilité. C'est en particulier vrai pour le cas de bibliothèques disponibles dans des outils, depuis plusieurs années, dont le re-développement dans un langage unifié conduit à des pertes de temps et de coûts considérables.

- La traduction des modèles dans un autre langage différent de leurs langages d'origine, où ils étaient initialement testés et validés, présente des risques de détérioration de robustesse. Une traduction exacte n'est pas toujours garantie et des erreurs ne peuvent pas être évitées même si cette traduction est effectuée automatiquement.
- Le recodage des modèles existant dans la bibliothèque d'un outil donné, suppose qu'ils sont ouverts. Ils doivent être « open source » exposant leurs variables, paramètres et équations pour qu'ils puissent être réutilisés, alors que ce n'est pas toujours le cas des outils de simulations du bâtiment. Pour le logiciel TRNSys, par exemple, bien que quelques uns de ses modèles soient documentés, d'autres ne le

sont pas. Seulement des DLLs correspondantes à ces composants sont accessibles à l'utilisateur [RIED 2009].

L'approche boîte blanche ne permet pas, **seule**, de répondre aux objectifs fixés en terme de modélisation systèmes :

- Elle ne permet pas de dépasser les limitations relatives aux capacités des simulateurs existants (cf chapitre 1): limitation à une phase d'étude et à un domaine scientifique. Elle ne permet pas d'améliorer les capacités des outils existants, leurs modularités, et en même temps propose un outil (environnement Modelica) avec des capacités limitées théoriquement et techniquement. Elle n'offre pas d'avantage de possibilités de collaborations avec d'autres outils dont on veut profiter des capacités de résolution.

- Elle ne permet pas non plus de résoudre complètement les problèmes relatifs aux hétérogénéités des modèles. Bien que plusieurs modèles puissent être développés dans Modelica, les modèles qui ne sont pas supportés par ce langage, comme un modèle multi-agent de l'occupant ou des formulations spécifiques d'équations aux dérivées partielles, ne peuvent pas être utilisés.

Ainsi, avec l'exploitation seule de cette approche boîte blanche, l'amélioration de la modélisation du système de PREDIS par l'intégration des nouveaux modèles plus détaillés (comme une modélisation fine de l'enveloppe réalisés par COMFIE ou une modélisation de l'occupant à l'aide de l'environnement BRAHMS), s'avère délicate.

Il est donc indispensable, dans l'objectif de palier à certaines de ces limitations, d'utiliser, en parallèle, une approche de modélisation système complémentaire. Il s'agit d'une approche qui se base cette fois sur le paradigme de composants logiciels et de « boîte noire ». Elle vient compléter la première approche grâce à deux principaux concepts : le plug-in et le plug-out (Figure 69) que nous définirons par la suite.

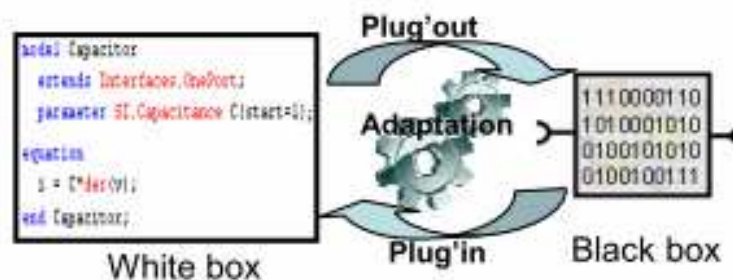


Figure 69. La complémentarité boîte noire/boîte blanche

III. L'approche boîte noire se basant sur le standard ICAr de composants logiciels

III.1. Le paradigme de composants logiciels

III.1.1. Le composant logiciel

Le concept de composant logiciel a vu son apparition dans l'ingénierie informatique au début des années 90 [SZY 1998] [MAU 2000] [BARB 2002]. Ce paradigme a été introduit après la programmation orientée objet pour dépasser ses limites en terme de décomposition et d'autonomie (Figure 70). Un composant logiciel est défini par Szyperski [SZY 1998] en tant que « une unité de composition qui est spécifiée, contractuellement, par ses interfaces et ses dépendances contextuelle explicites. Un composant logiciel peut être déployé indépendamment et est sujet de composition par des tierces entités. ». La figure suivante positionne ce paradigme entre celui des objets, bien adapté au couplage fort mais peu adaptable dynamiquement, et celui des services, offrant une souplesse de couplage plus grande mais des interactions plus limitées.

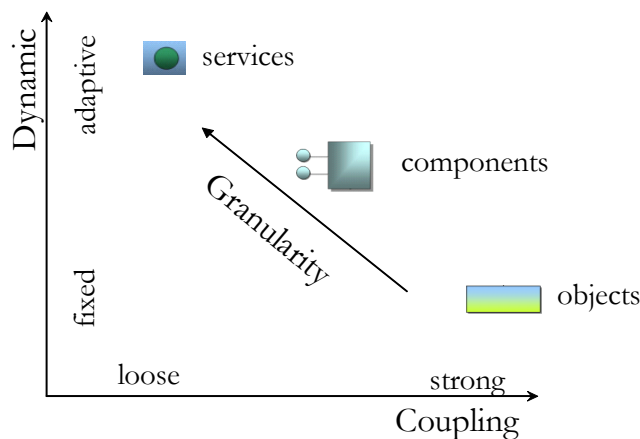


Figure 70. La souplesse des composants logiciels [DEL 2011]

Il s'agit donc :

- D'une entité autonome de déploiement, capable de s'exécuter indépendamment de l'outil de génération.
- D'une unité informatique encapsulant un code binaire, masquant son contenu pour préserver sa confidentialité et exposant uniquement ses interfaces et ses ports d'entrée/sortie (Figure 71).

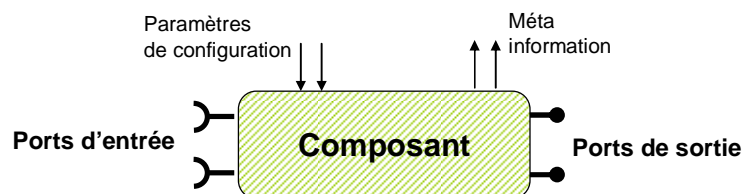


Figure 71. Vue externe d'un composant logiciel

- une entité de composition capable de se connecter à d'autres composants via ses ports de connexions [MEI 1997].

Les caractéristiques de ce paradigme de modélisation offrent un certain nombre d'avantages dont on peut citer :

- La modularité : des systèmes facilement décomposables et maintenables.
- La portabilité et la capitalisation : des modèles deviennent disponibles sur étagères et prêt à être utilisés.
- La facilité de réutilisation et de composition : gestion des communications externes (ports de connexion et interfaces de communication)
- La robustesse : des modèles déjà testés dans leurs outils de génération

Afin de renforcer les capacités d'interopérabilité et d'échange de modèles des outils existants de simulation du bâtiment, il devient indispensable d'améliorer leurs modularités [MAZ 2009]. Ces outils doivent permettre de facilement introduire des nouveaux modèles et, dans l'autre sens, d'extraire et d'utiliser leurs propres modèles à l'extérieur.

Un changement de paradigme était nécessaire pour assurer cette modularité et le composant logiciel a été adopté lors de développements de certains outils. TRNSys, par exemple, permet l'introduction de nouveaux modèles (fortran et C) qui seront ensuite compilés en bibliothèques dynamiques (DLL : code compilé) pour être utilisés en tant que TYPEs. Chaque modèle TYPE est ainsi défini par une DLL qui peut être exportée vers d'autres outils [RIED 2009].

Or, les composants pouvant être créés par plusieurs outils, sont caractérisés par une diversité syntaxique et sémantique rendant délicate leurs adoptions et interprétations dans d'autres environnements que celui d'origine. Ces différences peuvent ainsi limiter les collaborations entre outils et l'échange de modèles. D'où le besoin d'une standardisation de ces composants et de leurs interfaces de programmation dans le secteur du bâtiment.

III.1.2. Le standard de composants logiciels : Caractéristiques et concepts dérivants

III.1.2.1. Caractéristiques

Un standard de composant logiciel est défini par un patron (pattern) spécifiant ses ports de connexions, ses interfaces de communications ainsi que certaines méta information afin de normaliser la nature et le format des informations qui seront échangées par le composant. Une norme de composant logiciel doit ainsi spécifier ses trois dimensions (Figure 72):

-la **sémantique** et la **syntaxe** de ces modèles (interfaces de communications et ports de connexions).

-son emballage « **packaging** »: nécessaire pour faciliter sa distribution, le packaging spécifie la manière dont les informations du composant (modèles, meta-données, et autres ressources) sont embarquées au sein d'une unique entité de déploiement.

-ses mécanismes de réutilisation en **déploiement** et en **introspection** : comment déployer ce composant, puis l'interroger pour connaître les services disponibles.

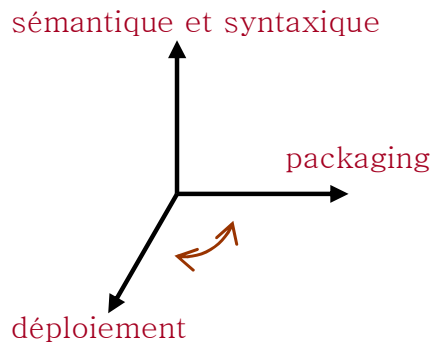


Figure 72. Les dimensions de la normalisation du composant logiciel

Une norme de composant logiciel devient un standard à partir du moment où de nombreux utilisateurs expriment les mêmes besoins et partagent les mêmes exigences [SZY 1996]. Le standard de composant se doit ainsi de fournir les services permettant de répondre aux besoins les plus larges de ce groupe de clients.

Comme dans plusieurs communautés, telle que l'automobile où le standard MODELISAR a été introduit, les utilisateurs de simulation du bâtiment ont exprimé le besoin de collaboration et de mise en place d'une norme commune adaptée (projets ANR SIMINTHEC et ANR Plumes). Pour cela, nous allons introduire le standard ICAR et l'adapter aux exigences de cette communauté.

Cette approche de standardisation va assurer un partage et une réutilisation plus facile des modèles caractérisés par des interfaces normalisées. Leurs import et export dans et depuis un outil donné peuvent ainsi se faire de manière automatique grâce à des programmes spécifiques (« plug-in » et « plug-out ».) et de manière générique pour n'importe quel composant de même type.

III.1.2.2. Les types de composants logiciels

On peut distinguer deux types de composants logiciels pouvant être générés par des plug-outs et utilisés via des plug-ins :

- Le composant-modèle : ce composant encapsule uniquement la description du modèle (équations) et nécessite l'utilisation d'un solveur externe pour effectuer son calcul et sa simulation. Il est ainsi capable de se connecter à d'autres modèles grâce à ses ports de connexion horizontale, mais également au solveur d'un outil externe grâce aux ports de connexion verticale.

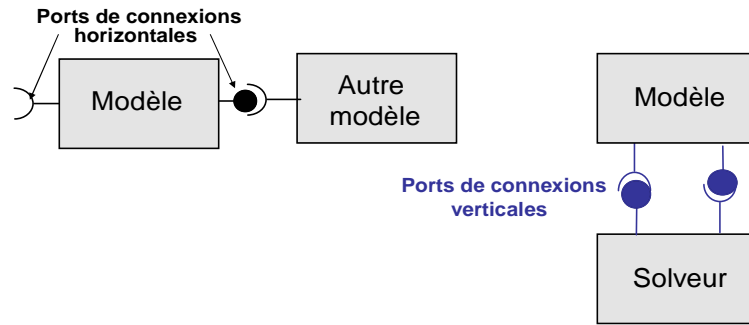


Figure 73. Le composant-modèle

○ Le composant-simulateur : il contient le modèle et son solveur associé (avec ou sans IHM) ou également son algorithme de résolution qui seront pilotés par le composant même. Il s'agit d'un composant autonome car il encapsule les moyens nécessaires à sa simulation. Cette autonomie n'est peut être pas absolue si le composant pilote un logiciel qui nécessite d'être déjà installé sur la machine (il y a alors dépendance via des ressources ce que savent gérer les approches composants, cf. par exemple OSGI [MAR 2001]).

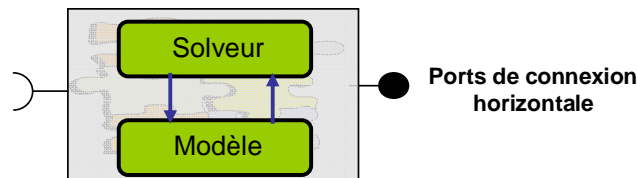


Figure 74. Le composant-simulateur

III.1.2.3. Un plug-out de composant logiciel

Un plug-out est un code informatique permettant d'exporter des fonctionnalités de modélisation/simulation depuis un outil de modélisation/simulation (Figure 75). Il peut être mis en place sans besoin de modifier le cœur de l'outil ou être en relation avec les développeurs de l'outil.

Un plug-out s'insère dans un framework global car le modèle ainsi exporté est destiné à apporter plus de fonctionnalités à un ou plusieurs logiciels existants. Et dans notre contexte où il permet de générer un composant logiciel dans une norme établie, il permet d'apporter plus de modularité à l'outil pour lequel il est développé et plus de portabilité pour ses modèles.

Un composant logiciel généré sous une norme spécifique peut être importé de manière **générique** dans n'importe quel outil dans lequel un plug-in de cette norme a été implémenté.

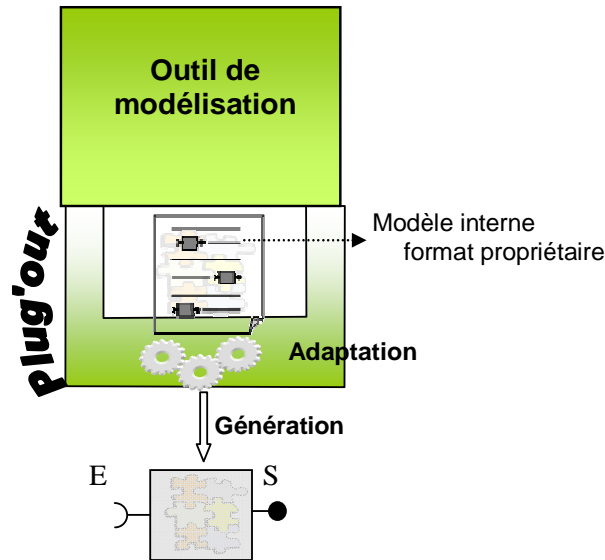


Figure 75. Le concept de plug-out

III.1.2.4. Un plug-in de composant logiciel

Ayant les mêmes caractéristiques qu'un plug-out, un plug-in représente un code informatique permettant d'importer des fonctionnalités de modélisation/simulation dans un simulateur/orchestracteur existant (Figure 76).

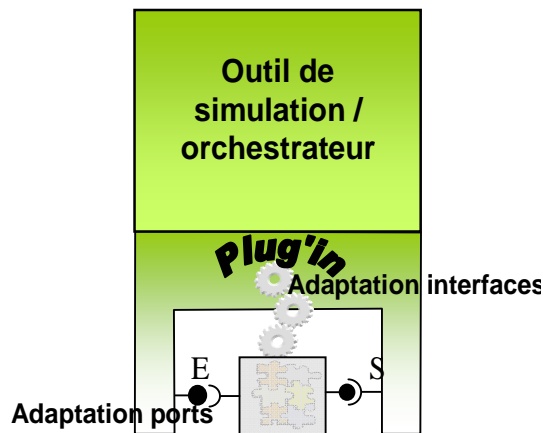


Figure 76. Le concept de plug-in

Dans notre contexte d'utilisation, il s'agit essentiellement d'une adaptation syntaxique entre les interfaces du composant importé et les API offertes par l'outil pour l'intégration d'un nouveau modèle (ou simulateur). La normalisation des interfaces de ces composants assure ainsi une utilisation automatisée grâce à des plug-ins génériques permettant d'importer tous les composants soumis au même standard.

III.1.2.5. Un bus à composants logiciels

Grâce à ces deux notions complémentaires, un bus à composants logiciels peut être créé. Il s'agit d'une structure informatique jouant le rôle d'intermédiaire entre plusieurs outils en assurant le partage des composants logiciels entre eux (Figure 77). Comme son équivalent

hardware, cette plate-forme doit permettre le déploiement, la configuration et l'instanciation des composants logiciels ainsi que la canalisation des communications avec les outils faisant la simulation et l'orchestration finale ainsi également qu'entre plusieurs composants. .

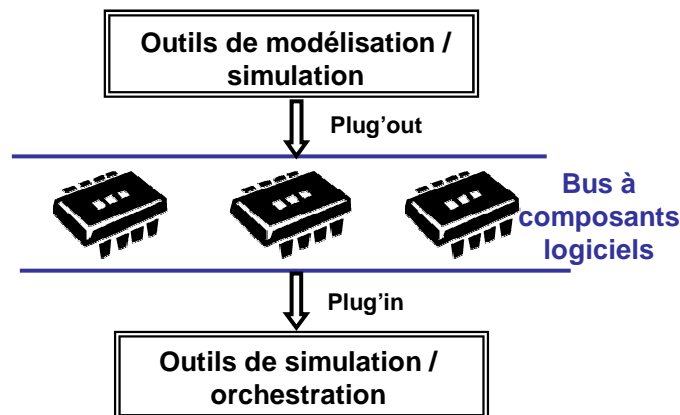


Figure 77. Le concept de bus à composants logiciels

III.2. Le standard « ICAR » de composants logiciels

Un standard de composant logiciel s'avère ainsi un puissant candidat capable de compléter l'approche d'interopérabilité boîte blanche basée sur le langage Modelica et améliorer les capacités d'interopérabilité des outils existants. Nous allons dans la suite introduire un standard de composants logiciels appelé « ICAR » en application à la modélisation et la simulation du bâtiment et montrer son adéquation potentielle aux exigences et aux outils de ce secteur.

La norme ICAR a été initialement développée pour des applications de génie électrique et pour répondre à certains besoins d'optimisation [DEL 2003] [FIS 2004]. Elle a connu par la suite plusieurs améliorations pour mieux correspondre à la simulation des systèmes mécatroniques lors du développement du logiciel RELUCTOOL [DUP 2006] [DO 2010] pour faire des composants intégrables dans le framework CADES [DEL 2012b]. Nous allons dans la suite présenter le pattern de l'ICAR, ses spécificités le positionnant comme un candidat potentiel pour la standardisation des modèles de la simulation dans le bâtiment en particulier.

III.2.1. Spécificités de la norme ICAR

L'ICAR représente un pattern spécifique de standardisation des composants logiciels sous forme d'un code Java compilé, défini par ses entrées et sorties, et les services qu'il fournit.

L'utilisation du Java comme langage de base de développement de l'ICAR (il peut contenir d'autres types de programmes) lui offre une bonne portabilité. Cela vient du fait qu'il s'agit d'un langage indépendant de la plateforme d'utilisation, contrairement aux langages C ou fortran. En effet, lors de la compilation du code Java, un code précompilé est généré (qui ne contient pas de code natif), puis il est interprété par une « machine virtuelle java » (JVM) avant d'être exécuté sur une machine spécifique..

En plus de l'unification de ses interfaces de communication et de ses ports de connexions étudiés dans le chapitre 4 (dimension syntaxique et sémantique), la norme ICAr spécifie également ses dimensions de packaging et de déploiement (Figure 72).

En effet, la norme de packaging de l'ICAr impose le format JAR du fichier généré. C'est en fait un fichier d'archives (*contenant plusieurs fichiers avec des fichiers meta-inf*) compressés (*format ZIP*). Ce fichier est exploitable par les API Java « `java.util.jar` » de la JDK. Concernant le déploiement, un outil de chargement est disponible (*codes.icar.io.ICArLoader*) et un point d'entrée normalisé « `component` » (Figure 78) pour permettre d'accéder aux classes Java contenues dans l'archive et ainsi de charger le composant en mémoire, l'interroger et le piloter via ses interfaces Java.

Ce packaging et ce déploiement sont compatibles avec la norme OSGI [MAR 2001]. Cette plate-forme permet le partage de services distants au niveau des réseaux et des dispositifs locaux. Cette technologie a été utilisée, par exemple dans la construction d'une architecture orientée services, dédiée aux bâtiments intelligents « `smart` » [WU 2007].

De plus, grâce à cette technologie Java, l'ICAr est également compatible avec les technologies des « `web services` ». Ainsi, des composants ICAr peuvent être disponibles sur des serveurs à distance et être utilisés comme si ils étaient présent localement [DEL 2012]. La mise à disposition de fonctionnalités de calcul, de simulation ou d'optimisation disponibles à distance dans des web-services appelée « `cloud computing` » peut offrir plusieurs avantages dont :

- Le développement durable : mutualisation des ressources de calcul sur des serveurs virtuels (économie sur le nombre de serveurs en fonctionnement), regroupés dans des data centre localisés dans des pays nordiques (économie sur le refroidissement).
- Sûreté de fonctionnement : services et stockage de données sur infrastructures fiables
- Disponibilité générale de services unitairement coûteux: licences de calcul, prévision/optimisation avec calculs lourds
- Possibilités de disposer de capacités de stockage et de calcul non limitées, au moins transitoirement.

Une des spécificités majeures de l'ICAr par rapport à d'autres standards, en particulier les « `fmi` » (chapitre 1, IV.1.2), est son architecture évolutive. Il est possible d'étendre les capacités du composant en lui ajoutant de nouvelles fonctionnalités telles que la simulation dynamique, l'optimisation et la documentation... et de découvrir dynamiquement les services offerts par le composant...En effet, l'ICAr peut être considéré comme un composant multiservices. Ces services sont disponibles via différentes facettes qui sont accessibles par l'intermédiaire de l'interface « `component` ». Cette interface représente un moyen d'accéder à ses ports de connexion (entrées/sorties) et interagir avec ses différents services ou « `facettes` » (Figure 78). Chaque facette du composant correspond à une tâche particulière capable de fournir les sorties désirées à parties des entrées fournies tel que le calcul du jacobien.

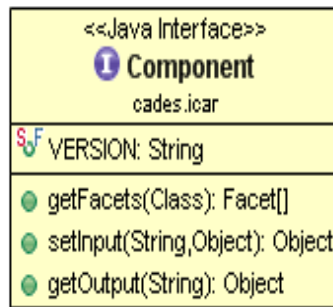


Figure 78. L'interface principale « component » de l'ICAr

III.2.2. Les facettes de l'ICAr

Plusieurs services peuvent être disponibles dans le composant « multi-facettes » comme le service d'optimisation [DEL 2003] et le service de simulation dynamique [DO 2010] (Figure 79). D'autres services peuvent être ajoutés selon l'évolution des besoins comme par exemple pour réaliser des stratégies de contrôle commande, du diagnostic ou du dimensionnement grâce à l'extensibilité de la norme ICAr et la modularité de ses facettes.



Figure 79. Le principe « multi-facettes » de l'ICAr

Ces services peuvent être exploités via leurs facettes correspondantes. A titre d'exemple, le service d'optimisation met en œuvre plusieurs facettes comme le calcul statique du modèle (« model solver ») et le calcul de jacobien.

Pour répondre aux besoins de modélisation dynamiques de plusieurs systèmes physiques, des facettes de simulation dynamique ont été définies dans les spécifications de l'ICAr. Ces facettes sont (Figure 80) :

- La facette ODE : pour la description des systèmes d'équations différentielles explicites discrets, continus et hybrides.
- La facette DAE : pour les mêmes système que précédemment, mais décrits sous une forme implicite. Cette facette est développée dans le cadre de la thèse d'Abir Rezgui [REZ 2012].
- La facette de Co-simulation, pour encapsuler et piloter des simulateurs. Il est possible de formaliser également les algorithmes de synchronisation mais ces travaux ne sont pas encore prévus.

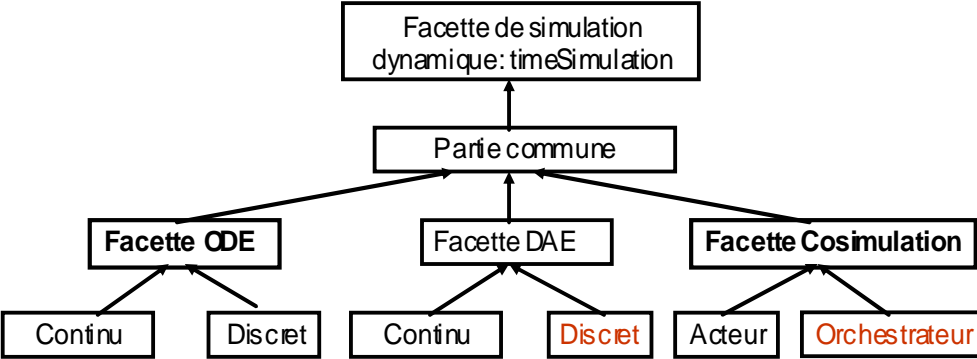


Figure 80. Les facettes dynamiques pouvant être contenues dans un ICAR et leur hiérarchie (les fonctionnalités en rouge ne sont pas encore intégrées)

III.3. Les plug-ins et les plug-outs autour du standard ICAR

On suppose qu'on veut réaliser la modélisation fine du système de PREDIS en suivant la procédure suivante : importer un modèle issu de COMFIE dans Dymola et le coupler au modèle de la VMC développé en modelica et détaillé précédemment. L'ensemble sera ensuite importé dans Matlab Simulink afin de tester différentes stratégies de contrôle commande et le coupler à un modèle d'occupant développé en Brahms (Figure 81). Afin de faciliter la réalisation de ce cas test, une architecture logicielle doit être mise en place et divers plug-ins et plug-outs du composant ICAR doivent être développés. Il s'agit donc de disposer du plug-out COMFIE, plug-in Dymola, Plug-out Dymola, Plug-in Matlab et Plug-out Brahms.

Nous détaillons maintenant les développements et l'utilisation de ces plug-ins et plug-outs réalisés dans cette thèse et utilisés dans notre cas test de simulation de PREDIS.

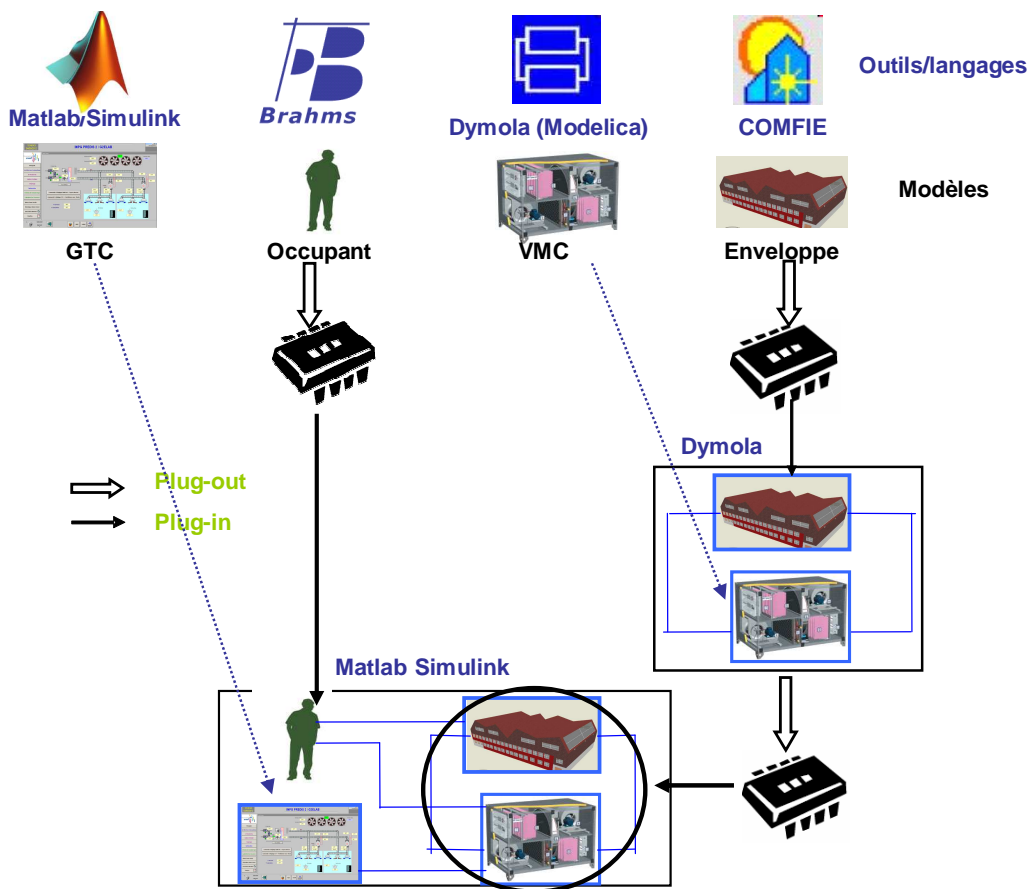


Figure 81. Exemple d'utilisation des plug-ins et des plug-outs de l'ICAr : réalisation d'un processus de simulation complet appliqué à la plate-forme PREDIS

III.3.1. Les plug-ins du standard ICAR

III.3.1.1. Généralités sur la mise en œuvre

III.3.1.1.1. Les principales fonctionnalités du plug-in

Un plug-in doit principalement assurer les fonctionnalités suivantes (Figure 82):

- L'adaptation syntaxique des interfaces de communications : étape nécessaire pour l'exécution du modèle. Cette étape est générique et indépendante du composant

logiciel utilisé, pour une même facette implémentée. Un « template »²² d'adaptation entre les interfaces du composant et les APIs d'un nouveau modèle de l'outil est généralement suffisant. Cette adaptation va permettre l'exécution de l'interface du composant correspondante à l'API du modèle appelé (par le solveur ou l'orchestrateur de l'outil).

- L'adaptation dynamique des ports de connexion propres à chaque composant : Selon les possibilités offertes par l'environnement hôte du plug-in, cette fonctionnalité peut être gérée de manière générique (tel est le cas du plug-in Matlab) ou non. Dans la négative, il est possible de proposer un programme complémentaire afin de générer les connexions du modèle au cas par cas. C'est ce que nous avons par exemple dû faire pour le plug-in Dymola.

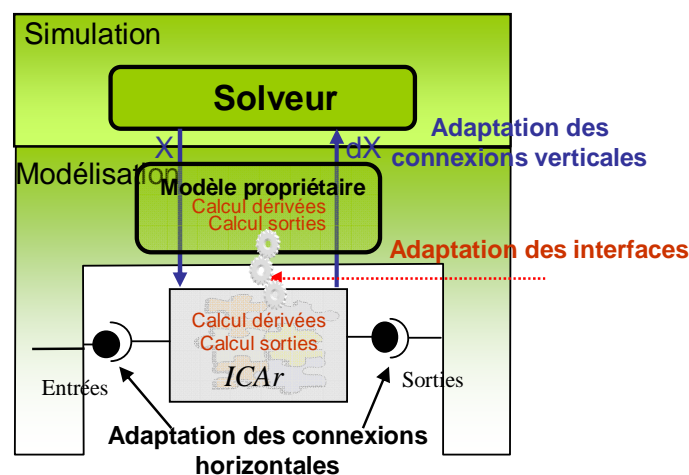


Figure 82. Illustration des principales fonctionnalités d'un plug-in de la facette ODE de l'ICAr

III.3.1.1.2. Les phases de vie d'un plug-in

La mise en place d'un plug-in est soumise à trois principales phases:

- une phase de **développement** « design time »: cette phase concerne les développeurs du plug-in qui n'ont pas nécessairement une relation directe avec les développeurs de l'outil. Elle consiste à mettre en place les programmes nécessaires au bon déroulement des autres phases : au déploiement et à l'exécution des plug-ins chez les utilisateurs.
- une phase de **déploiement** « deployment time »: les difficultés de cette phase résident dans le fait de rendre un plug-in autonome et facile à installer chez n'importe quel utilisateur. Elle consiste essentiellement dans le packaging des bibliothèques nécessaires (les bibliothèques de l'ICAr comme le kit de développement, les dlls d'adaptation avec le langage C...) et la gestion de variables d'environnement si nécessaires...

²² Modèle de conception de logiciel ou de présentation des données.

- une phase d'**utilisation** « run time »: dans cette phase, on exploite le plug-in installé et les développements réalisés pour profiter des nouvelles fonctionnalités apportées à l'outil.

Bien que la phase de développement d'un plug-in puisse s'avérer complexe et difficile à mettre en œuvre, son installation devient simple et son exploitation facile pour un utilisateur externe grâce au travail réalisé dans la phase de développement. Ces aspects génériques pour tous les plug-ins (et plug-outs) seront illustrés à travers le plug-in Matlab.

III.3.1.2. Le plug-in Matlab / Simulink

Le point de vue du développeur

Pour utiliser un composant ICAR (dynamique) dans Matlab/Simulink, il est nécessaire d'exploiter l'interface de S-Function [WEB_SFU], spécifiquement les M S-fonction (en Matlab). Matlab présente l'avantage d'incorporer sa propre machine virtuelle Java (JVM) ce qui facilite l'utilisation des fonctionnalités Java dans le langage Matlab (Figure 83).

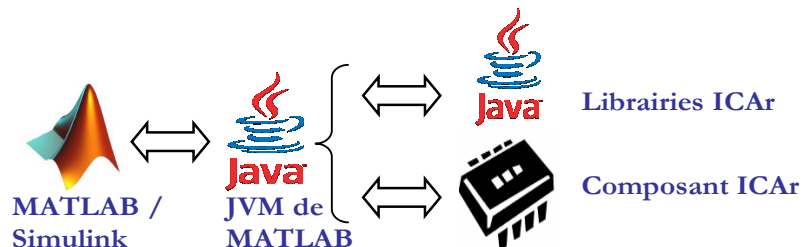


Figure 83. Le plug-in Matlab/Simulink

Le développement du plug-in Matlab/Simulink à nécessité deux principales étapes :

- Le développement d'un script M-File : permettant le chargement et l'introspection du composant afin d'identifier ses ports de connexions et les facettes disponibles (ODE pour la simulation dynamique). Des noms de variables génériques sont attribués (aux entrées/sorties, variables d'états...) et dont les valeurs se mettent à jour automatiquement depuis l'espace du travail avec le lancement du script d'introspection.
- Le développement d'une « S-fonction » (modèle Simulink) : réalisant l'adaptation syntaxique des interfaces de communication entre Simulink et l'ICAR. Ce modèle simulink est générique et peut ensuite être utilisé pour n'importe quel modèle ICAR. En effet, cette « S-fonction » est capable de dynamiquement définir ses ports génériques grâce à l'introspection de l'ICAR.

Le point de vue de l'utilisateur

Pour l'utilisateur, on peut distinguer deux phases, l'installation et l'utilisation du plug-in. L'installation nécessite un démarrage particulier de Matlab afin de spécifier les librairies Java nécessaire à l'utilisation des ICAR. L'utilisation exploite ensuite deux fichiers, le premier étant un script Matlab à exécuter en lui spécifiant le chemin du fichier ICAR à exploiter, le

second correspondant à la S-Fonction générique (s'adaptant automatiquement à l'ICAr chargé) à insérer simplement dans Simulink.

A la simulation la S-fonction va jouer le rôle d'intermédiaire entre le composant ICAr et le solveur de Simulink et va permettre, à chaque étape et à chaque requête (Annexe 7), l'exécution de l'interface correspondante de l'ICAr (cf chapitre 4) de manière complètement transparente pour l'utilisateur.

Le Plug-in ainsi que les éléments nécessaires à son exécution sont disponibles en téléchargement sur le site Dimocode [WEB_PMAT]. Le tutorial fourni permet la prise en main du programme [SIM 2010].

III.3.1.3. Le plug-in de l'ICAr vers Dymola

Le plug-in Dymola consiste à importer un composant logiciel ICAr et à l'utiliser en tant que modèle Modelica pouvant être connecté à d'autres modèles et simulé dans cet environnement (Figure 84).

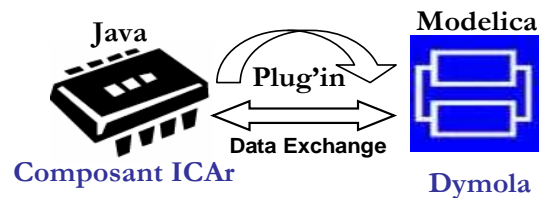


Figure 84. Le plug-in Dymola

Pour pouvoir importer dans un environnement Modelica des modèles décrits dans un autre langage (C ou JAVA), il est possible d'utiliser la fonctionnalité « external » [SPEC 2010].

Ainsi, l'interfaçage entre le composant ICAr en Java et un modèle Modelica peut être réalisé en passant par les fonctions « externes » en Java [SJO 2009a] [OLS 2005]. Cette solution passe en réalité par un code C généré automatiquement pour s'interfacer avec le code Java [SJO 2009b]. En vue de réduire les risques liés à l'interfaçage automatique C->Java, nous avons préféré travailler nous même en langage C. Le plug-in réalisé se chargera alors du pilotage des composants ICAr en Java via la technologie de JNI [WEB_JNI] (Figure 85).

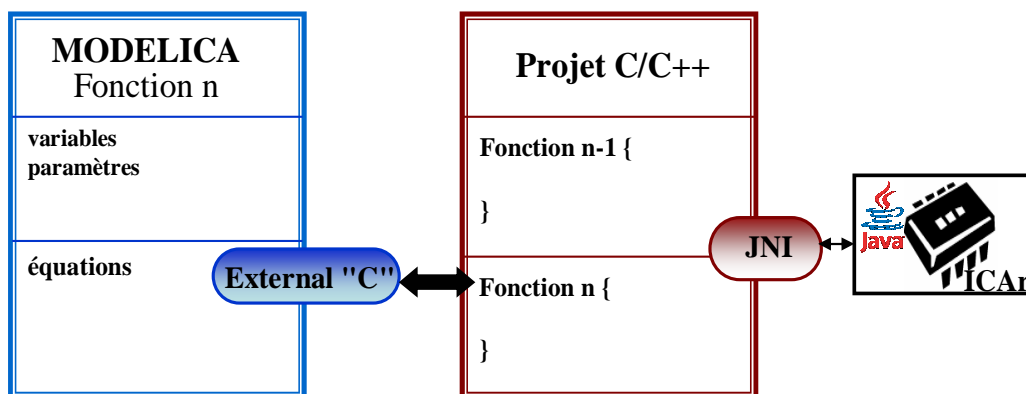


Figure 85. Les étapes du plug-in Dymola

La réalisation de ce plug-in se décompose en trois principales parties (Annexe 8)

- L'interfaçage entre les langages Java et C via la JNI.
- L'interface entre les langages C et Modelica via la fonctionnalité « external ».
- Un outil de génération automatique du code Modelica spécifique à chaque ICAR, permettant de spécifier explicitement le nom et nombre des ports (Figure 86). Ce programme sera exécuté par l'utilisateur pour chaque nouvel ICAR.

```

model ICARCOMFIEODE
  parameter String path= "../ICARCOMFIE.icar";
  parameter Real P1=3500;
  parameter Real P2=0;
  Real[2] outputs;
  String[2] outputsNames={"T1","T2"};
  Real[16] stateVariables;
  String[16] stateVariablesNames={"X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","X11","X12","X13","X14","X15","X16"};
protected
  constant Integer icar_stateVariableNumber=16;
   Package des fonctions externes
initial equation
  IcarFunctions.Initialize(path);
  stateVariables=IcarFunctions.initialStateVariables_Retrieve(icar_stateVariableNumber);
  IcarFunctions.setScalarInputs({"P1","P2"},{P1,P2});
equation
  IcarFunctions.setStateVariableInputs(stateVariablesNames,stateVariables);
  {outputs}=IcarFunctions.scalarOutputsComp_Retrieve(time,outputsNames);
  {der(stateVariables)}=IcarFunctions.stateVariableDerivatives_Retrieve(time,icar_stateVariableNumber);
end ICARCOMFIEODE;
  
```

Figure 86. Exemple du code Modelica automatiquement généré (ICAR ODE de COMFIE)

III.3.2. Les plug-outs vers le standard ICAR

III.3.2.1. Généralités

Un plug-out de composant ICAR est un programme informatique dont la mise en place passe également par les trois phases de développement, déploiement et utilisation. Il est généralement réalisé en Java (langage de développement de l'ICAR) qui se base sur le kit de développement fourni avec l'ICAR pour y implémenter une facette et un service donné. Ce développement consiste à définir, à partir des variables d'un modèle (modèle seul ou modèle +solveur) issu d'un outil donné, les entrées/sorties de l'ICAR (ports de connexions) et compléter les interfaces du composant par les équations ou les API correspondantes du modèle à exporter.

L'étape d'adaptation des interfaces entre le modèle et le composant ICAR peut nécessiter une adaptation sémantique (modification de formalisme par exemple) en plus de l'adaptation syntaxique classique. Cette étape intervient pour permettre de décrire le modèle dans le formalisme imposé par la facette de l'ICAR en question (ODE, DAE) quand il est différent de celui de son origine et nécessite un traitement beaucoup plus conséquent que l'adaptation syntaxique (eg. DAE vers ODE [VER 2012]).

A l'issu de son exécution, ce programme permet de générer un composant logiciel ICAR, indépendant de son outil d'origine et peut être utilisé dans plusieurs autres outils grâce aux plug-ins mis en place.

III.3.2.2. Le plug-out Modelica

Ce plug-out a été réalisé par Franck Verdière (Ingénieur de recherche au G2ELab) où deux alternatives ont été testées, directement à partir du langage Modelica ou via un composant de simulation FMI.

La première perspective consiste à exploiter la capacité de plusieurs outils à générer un composant FMI pour l'encapsuler dans un composant ICAR. Il s'agit uniquement d'une adaptation syntaxique (même formalisme ODE) des ports et des interfaces entre les deux composants. Il est donc indispensable de passer par la JNI pour assurer la communication entre les deux langages (C des FMI et java des ICAR). Cette solution fait perdre à l'ICAR ses avantages de portabilité à cause du FMI qu'il embarque.

La deuxième méthode s'appuie sur le programme open source JModelica. Le principe de ce plug-out consiste à intervenir, au niveau du processus de transformation effectué par le compilateur modelica, à l'étape de génération et une fois que l'analyse du code est effectuée [AKES 2009], pour modifier le code à générer. Un code Java sera ainsi généré à la place du code C classique (Figure 87). Pour créer un composant logiciel implémentant la facette ODE, ce code est adapté syntaxiquement et sémantiquement au standard ICAR [VER 2012].

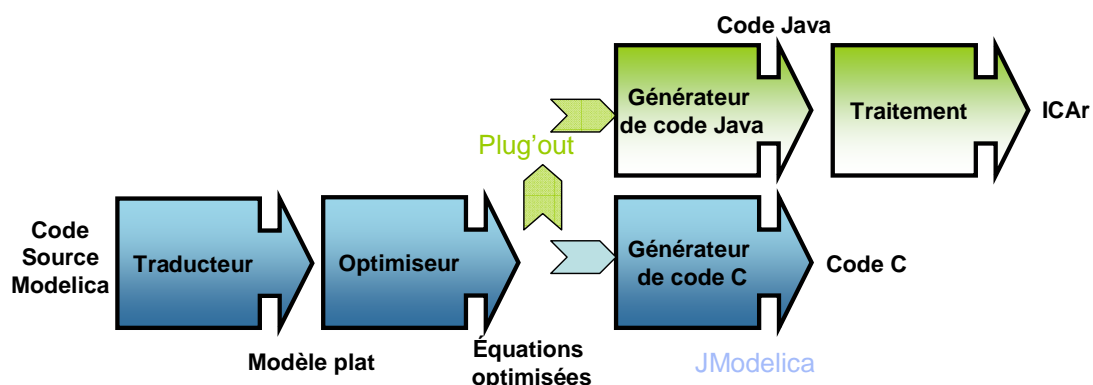


Figure 87. Le principe du plug-out d'un modèle Modelica (via JModelica)

Ainsi, nous disposons d'un outil nous permettant d'exploiter des modèles Modelica dans d'autres environnements tels que Matlab/Simulink via les composants logiciels ICAR.

III.3.2.3. Le plug-out de Brahms

L'outil Brahms ne permet pas une séparation entre le modèle et son solveur (Machine virtuelle) car ses modèles multi-agents nécessitent un traitement particulier. Seul un composant-simulateur (la facette de co-simulation) peut être construit à partir de Brahms.

En outre, Brahms possède l'avantage d'offrir des API en Java permettant de piloter sa machine virtuelle (l'équivalent d'un solveur). Donc l'étape principale du développement de ce plug-out consiste à l'adaptation syntaxique entre les interfaces de l'ICAR et les API correspondantes de Brahms (Figure 88).

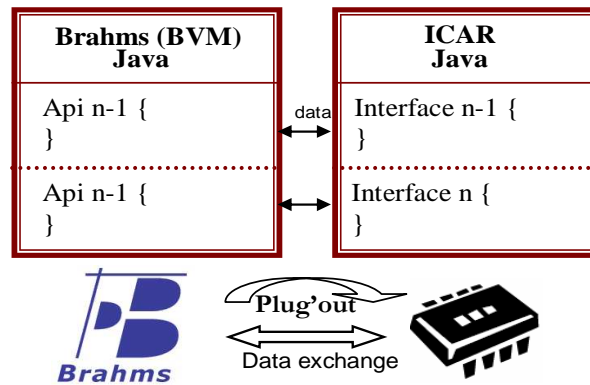


Figure 88. Le principe du plug-out de Brahms

III.3.2.4. Le plug-out depuis COMFIE

La spécificité de ce plug-out, détaillé en annexe (Annexe 9) [GAA 2012], est la possibilité d'implémenter plusieurs types de facettes selon les objectifs d'utilisation du composant: ODE continu et discrète, et facette de co-simulation. Plusieurs versions du système d'états peuvent être reconstruites pour alimenter les différentes interfaces de l'ICAR (Sys 2 et Sys 3 du chapitre 2). Une adaptation sémantique est donc nécessaire pour assurer une compatibilité des formalismes de description. Un exemple d'adaptation au formalisme ODE est détaillé dans le chapitre 4.

III.3.3. Limites et perspectives d'améliorations des plug-ins et des plug-outs

Plusieurs plug-ins et plug-outs de la norme ICAR ont été développés. Certains ont été détaillés dans ce chapitre car en lien avec des applicatifs étudiés dans cette thèse et d'autres moins comme ; un plug-out depuis l'environnement CADES [DEL 2003] ou le plug-in dans TRNSYS (Annexe 10) ou encore le plug-in VHDL-AMS [REZ 2011] (Figure 89)... D'autres sont en perspectives comme un plug-out depuis TRNSYS ou également un plug-in vers l'environnement de co-simulation BCVTB pour pouvoir profiter des développements réalisés dans cette plateforme. Ces travaux ont pour objectifs de développer et d'enrichir le bus à composants logiciels afin de couvrir le maximum d'outils et mieux valoriser cette approche d'interopérabilité.

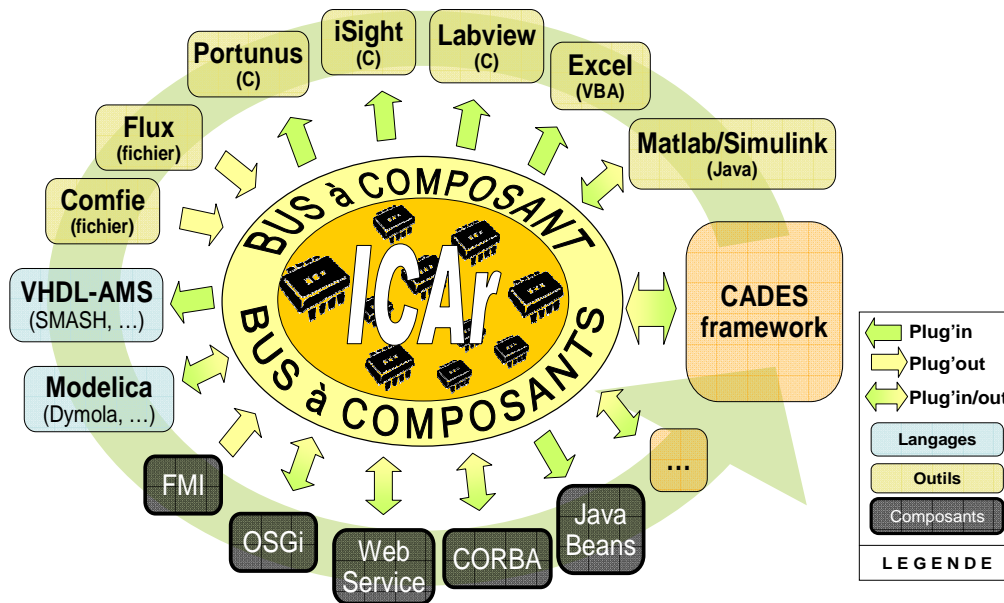


Figure 89. Le bus à composant logiciel développé autour de l'ICAr [DEL 2012C]

Malgré plusieurs développements, certaines limites sont repérées dans ces programmes à différentes phases :

- Au niveau de la phase de développement : - Pour plusieurs de ces plug-ins et plug-outs, certaines facettes uniquement ont été développées selon les besoins des cas tests sélectionnés (chapitre 4). D'autres facettes ne sont pas encore traitées ou parfois simplement impossibles à réaliser (limitations des outils). - Pour un seul outil, on peut avoir différents programmes de plug-in et de plug-out spécifiques à chaque facette. Il est donc nécessaire d'automatiser la sélection du programme à utiliser selon le type du composant.

- Au niveau de la phase de déploiement : Bien que certains plug-ins et plug-outs soient rendu autonomes et facilement « déployables » chez les utilisateurs, comme TRNSYS et Matlab, disponibles sur le site collaboratif DIMOCODE [WEB_DIM], d'autres le sont moins. Ce travail d'amélioration est prévu dans une phase plus avancée d'industrialisation des outils par la société Vesta-System, valorisant en particulier le framework CADES [WEB_VES]. On s'est essentiellement focalisé, dans ce travail, à montrer la faisabilité de cette technique et l'intérêt de l'approche adoptée.

- Au niveau de la phase d'utilisation : - la génération automatique des modèles et la création automatique des composants logiciels restent partiellement traitées dans certains plug-ins et plug-outs. On peut citer le manque d'automatisation lors de l'adaptation des ports entre l'ICAr et le Type TRNSYS, la non prise en compte de certains cas spéciaux dans COMFIE (zone mitoyenne, contact direct avec le sol...). Ces problèmes peuvent être facilement traités et corrigés. - En outre des problèmes de mémoire lors de l'utilisation des composants ont été détectés ainsi que certaines dégradations des performances de calcul induit, en particulier, par les passages C/Java. Ces outils seront exploités essentiellement pour la validation et la vérification des choix de la phase de conception et les performances de cette

technique y restent ainsi acceptables compte tenu des autres avantages qu'elle apporte en terme d'interopérabilité.

III.4. Synthèse sur la complémentarité des deux approches d'interopérabilité

La technique se basant sur un standard de composant logiciel s'avère ainsi une solution pertinente permettant de dépasser les principales limites du langage Modelica. Les problèmes théoriques, techniques et pratiques de ce langage peuvent être résolus grâce à la notion de plug-in vers ce langage. Il constitue ainsi une extension du langage et lui apporte des capacités supplémentaires en terme de modélisation et de simulation (différences finis, multi-agents). Des nouveaux modèles et simulateurs peuvent être introduits facilement, de manière automatique et robuste, ce qui permet l'amélioration des capacités d'interopérabilité et d'accueil de composants externes.

Cette technique n'est intéressante que si dans l'autre sens, plusieurs plug-outs de divers outils et langages sont développés de manière à construire un bus à composant logiciels. C'est dans cet objectif que plusieurs plug-outs ont été réalisés et doit continuer de s'étendre avec la mise à disposition des normes et de tutoriaux de mise en œuvre.

Il est en même temps intéressant de pouvoir profiter des facilités et des capacités de modélisation de ce langage en exportant ces modèles vers d'autres outils pour dépasser leurs limites. Grâce au plug-out Modelica, il devient donc possible d'exporter facilement ces modèles. Le modèle Modelica encapsulé dans un composant logiciel gagne ainsi en autonomie et en portabilité lui offrant la possibilité d'utilisation dans d'autres outils où des plug-ins ont été développés. D'où la complémentarité entre les deux concepts. Ce modèle peut être aussi transformé en un service web et utilisé à distance [DEL 2012].

Grâce au concept de plug-in, un composant logiciel issu d'un autre outil, encapsulant ses connaissances de modélisation et ses capacités de calcul, peut être importé dans Modelica pour le compléter. Cette complémentarité est non seulement matérielle mais aussi conceptuelle. Un modèle boîte noire (composant) peut être couplé à un modèle boîte blanche (Modelica) formant ainsi un modèle boîte grise [JON 2007]. Le modèle « gris » expose certaines parties de ses équations (ce qu'on veut monter) et éclipsent d'autres (confidentielles ou non disponibles).

Un composant logiciel peut être généré à partir de cette boîte grise, grâce au plug-out Modelica, pour générer un composant logiciel composite (Figure 90) pouvant être importé dans plusieurs outils comme Matlab/Simulink.

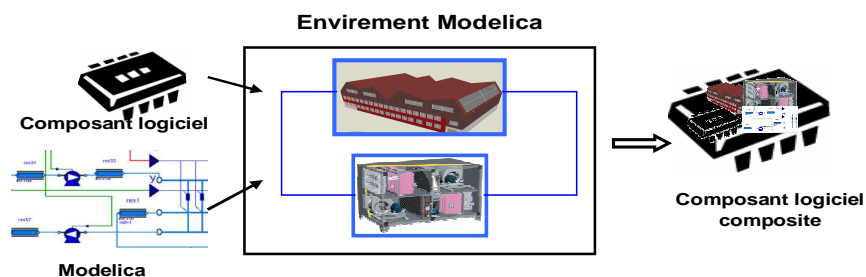


Figure 90. Le principe de composant composite

IV. Conclusion

Dans ce chapitre, nous avons étudié en application à des cas test tirés du bâtiment PREDIS, l'intérêt d'une approche de modélisation système se basant sur le langage multi-physique émergent Modelica. Les caractéristiques de ce langage ainsi que les limites de cette technique ont été évoquées.

Une autre approche boîte noire se basant sur le standard ICAr de composant logiciel a été également proposée pour compléter la première approche. L'exploitation de cette approche a pour vocation l'amélioration des capacités de modélisation, simulation et interopérabilité du langage Modelica et de ses environnements (Dymola spécifiquement). Deux principaux concepts dérivent de cette approche : les plug-ins et les plug-outs que nous avons définis et détaillés à travers des exemples de développements.

Bien que plusieurs exemples de plug-ins et de plug-outs aient été détaillés dans ce chapitre, nous n'avons pas illustré en application à des exemples concrets leurs utilités et leurs apports à l'approche classique de modélisation boîte blanche. Dans le chapitre suivant, des cas tests seront étudiés pour illustrer l'approche par composant logiciel et le standard ICAr en application au bâtiment.

Nous avons, dans ce chapitre, décrits les développements réalisés pour la mise en œuvre de certains plug-ins et plug-outs d'un point de vue extérieur sans entrer dans les détails des sémantiques des interfaces de communications et de formalisation des ports de connexions. Ces divers points seront abordés dans le chapitre suivant où ils seront étudiés pour des exemples spécifiques et généralisés pour couvrir plusieurs cas. Pour donner une vision de l'ICAr, ses caractéristiques et ses capacités, il est indispensable d'aborder les détails techniques de la structure de l'ICAr et les détails en terme de facettes et de ports de connexions.

Chapitre 4 :

La simulation dynamique des systèmes hétérogènes basée sur l'approche à composants logiciels

I. Introduction

Dans le chapitre précédent, on s'est intéressé à la modélisation des systèmes hétérogènes et les moyens mis en œuvre pour assurer une modélisation complète du bâtiment tenant compte des particularités de chacun de ses acteurs à travers deux solutions complémentaires : la première se base sur une approche boîte blanche utilisant principalement le langage Modelica et la seconde se base sur une approche boîte noire sur le concept de composants logiciels.

Dans ce chapitre, on va se focaliser sur l'approche basée sur le standard ICAr de composant logiciel. On va procéder, à travers des cas d'application concrets, à l'étude des fonctionnalités et des interfaces devant être implémentées dans ces composants génériques. Le composant se doit de pouvoir encapsuler plusieurs types de modèles et d'assurer une simulation efficace du système dans son ensemble et une bonne gestion de l'hétérogénéité de ses composantes.

Afin d'identifier et de spécifier ces interfaces, l'orchestration de ces systèmes hétérogènes sera étudiée sur deux niveaux.

- Au niveau de l'orchestrateur ou le solveur, afin d'identifier des stratégies efficaces de gestion de cette hétérogénéité et conclure sur les interfaces des modèles (modèle seul ou modèle et son solveur).
- Au niveau des modèles orchestrés, afin d'identifier la sémantique et la syntaxe des interfaces permettant de répondre ou besoins de modélisation et de ces stratégies d'orchestration.

Deux principales techniques d'orchestration des systèmes hybrides et de natures dissemblables seront abordées: la simulation hybride à pas variable dans la première partie et la co-simulation multi-cadences (« multi rate ») dans la deuxième. Les intérêts de ces techniques dans la gestion des disparités des modèles seront ainsi montrés

En plus de cet aspect scientifique de création des interfaces de composant, un aspect technique sera également élaboré le long de ce chapitre. Il s'agit de l'architecture logicielle mise en place pour faciliter, automatiser et généraliser ces stratégies d'orchestration. Le processus d'exploitation de certains plug-ins et plug-outs développés (cf chapitre 3) sera illustré en application aux exemples traités pour l'aspect sémantique.

Nous allons donc traiter des problèmes d'interopérabilité des modèles et des simulateurs à travers des cas test spécifiques. Mais les techniques et concepts exploités seront généralisés à des plus larges utilisations.

II. La simulation à pas variable des systèmes hybrides

II.1.Introduction

Définition de la simulation à pas variable

La simulation à pas variable est une technique adaptée pour la simulation des systèmes hybrides. Un système hybride est un système contenant à la fois des variables d'état continues et des variables discontinues ou/et des événements. En effet, une simulation efficace de ce type de système devrait garantir un compromis entre le temps de calcul et la précision et devrait être capable de gérer les diverses discontinuités tout en assurant la continuité des variables d'état pour chaque changement d'état des variables discrètes [DO 2010]. Ainsi, pour améliorer les performances de la simulation sans effectuer de nombreux calculs supplémentaires, l'adaptation du pas de calcul selon la dynamique du système s'avère une solution pertinente. D'où l'appellation de simulation hybride à pas variable.

Les intérêts de la simulation hybride à base de composants logiciels

En plus des intérêts apportés par variation du pas de temps à la simulation des systèmes hybrides, l'application de l'approche composants logiciels va assurer une standardisation, une capitalisation des modèles, ainsi qu'une automatisation et une généralisation de la procédure de mise en place de cette simulation.

En effet, l'établissement d'un standard de composants logiciels pour la simulation hybride va permettre, principalement, l'unification des interfaces de ces composants les rendant ainsi facilement exploitables par plusieurs outils et leurs solveurs, permettant ainsi une simulation via diverses techniques. Leurs utilisations peuvent ainsi se faire de manière automatique, et deviennent génériques grâce aux plug-ins développés permettant d'importer tout modèle ICAR.

Afin de mieux généraliser ce concept, les spécifications du standard de composant doivent être définis de manière à permettre l'encapsulation de différents types de modèles : continus, discrets et hybrides indépendamment de leur outil d'origine. C'est dans cette perspective qu'a été établi la norme ICAR, qui grâce à son aspect multi-facette permet de supporter plusieurs types de modèles avec leurs interfaces associées. Les ports et les interfaces ont été également établis après études des plusieurs implémentations offertes par les outils.

Nous allons donc, dans cette partie du chapitre, monter à la fois l'intérêt de la simulation à pas variable, l'aspect générique du composant ICAR en mettant en valeur sa sémantique permettant de satisfaire aux besoins de ce type de simulation, ainsi que les avantages de son utilisation.

II.2. Architecture d'un cas d'étude : couplage entre un modèle d'enveloppe thermique (issu de COMFIE) et un modèle de chauffage (Matlab)

Le système étudié est composé d'un modèle thermique représentant le comportement de l'enveloppe d'un bâtiment, couplé à un modèle de chauffage électriquement commandé par une régulation à hystérésis selon la température de la pièce [GAA 2010]. Le bâtiment est composé de deux zones thermiques dont une seule est supposée être équipée d'un chauffage et qui sera étudiée par la suite. A chaque pas de simulation, les deux modèles vont interagir et échanger les informations concernant la température de la part du modèle thermique et la puissance électrique engagée de la part du modèle de l'ensemble chauffage et thermostat (Figure 91). Selon la température calculée par le modèle thermique et perçue par le thermostat, ce dernier va se décider à engager ou non la puissance électrique nécessaire pour son fonctionnement selon l'atteinte ou non des seuils de température. Cette puissance électrique va influencer l'évolution des températures de la zone où est installé le chauffage.

Les cas test spécifiés ci-dessous seront simulés et étudiés dans l'environnement Matlab/Simulink. MATLAB/Simulink, largement utilisé dans le domaine du contrôle commande, a été sélectionné ici pour ses performances en calcul matriciel, sa facilité de prototypage, ainsi que pour les capacités de ses solveurs en terme de prise en compte des systèmes hybrides.

Le modèle du chauffage est un modèle à événements qui est construit directement à partir des éléments de la bibliothèque standard de Simulink. Le modèle de l'enveloppe thermique est établi par reconstruction du système d'état décrivant le comportement de l'enveloppe. Ce système d'état est défini à partir des matrices d'états générées par COMFIE à l'issue d'une simulation du bâtiment étudié (cf chapitre 2 III.1). Le système d'état sélectionné dépend du type du modèle désiré (continu, discret) et peut être utilisé dans MATLAB/Simulink :

- directement par reconstruction manuelle d'une S-fonction correspondante
- indirectement en passant par les composants logiciels et de manière automatique grâce au plug-in dédié.

La difficulté de ce couplage réside dans l'hétérogénéité des natures temporelles des deux modèles composants ce système hybride (modèle continu ou discret et modèle à événement). Cette hétérogénéité induit des problématiques de gestion d'événements et de gestion du pas de simulation. La technique de simulation à pas fixe (pas d'1 heure puis pas 6 minutes) sera testée dans un premier temps, pour conclure sur ses limites en application à ce type d'étude. L'adoption d'une simulation à pas variable sera ensuite étudiée, nécessaire pour mieux gérer ce système hybride.

Afin de comparer les performances de simulation des techniques employées dans cette partie (pas fixe et pas variable, avec et sans composants logiciels), la même configuration a été préservée : mêmes données météorologiques (température extérieure et flux solaire),

puissances internes nulles (sauf celle due au chauffage) et même période de simulation (deux semaines). Le type du solveur varie selon les besoins en terme de simulation.

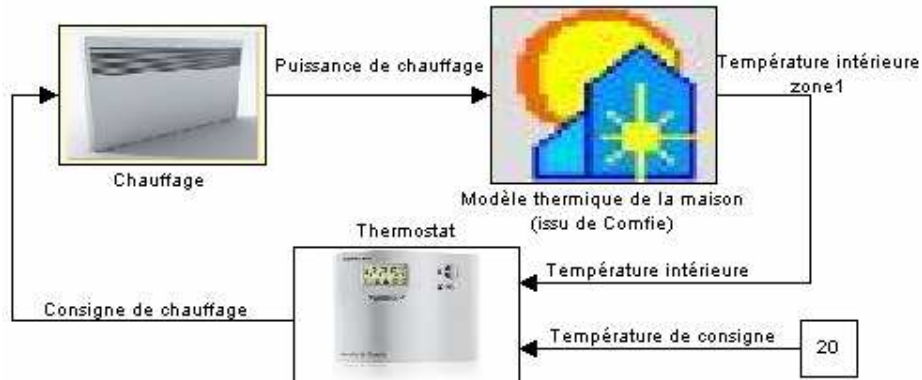


Figure 91. Schéma global de la simulation dans Simulnk

Ces problèmes d'interopérabilité seront donc traités via le concept de composant logiciel. La norme ICAr a été définie de manière à être capable d'adopter plusieurs types de modèles (continus, discrets et événementiels) et être compatible avec diverses stratégies de simulation (pas fixe et pas variable). Ces caractéristiques vont favoriser son aspect générique pour couvrir plusieurs cas test.

II.3.L'orchestration des modèles

II.3.1. La simulation à pas fixe

II.3.1.1. La simulation à pas fixe d'une heure

Dans cette partie, le block de S-fonction correspondant au modèle thermique est construit à partir des matrices d'états du système discret (chapitre 2, Sys 3 rappelé ci dessous), permettant de réaliser une simulation à pas fixe. Le modèle du chauffage est assimilé, ici, à une puissance constante.

$$\begin{cases} X^{n+1} = \exp(F \cdot \Delta t) \cdot X^n + W^\alpha \cdot (U^{n+1} - U^n) + W^\beta \cdot (Y^{n+1} - Y^n) \\ Y^{n+1} = MGIF \cdot X^n + MGIE \cdot U^{n+1} + MGID \cdot U^n - MGIC \cdot Y^n \end{cases}$$

Dans un premier temps, le pas de simulation a été fixé à une heure, imposé par le système discret de l'enveloppe défini pour un seul pas de temps (la matrice d'état dépend du pas de temps). Ce pas de temps est suffisant pour simuler le comportement de l'enveloppe qui est caractérisée par des constantes de temps élevées dues à son inertie thermique pouvant atteindre plusieurs heures dans le cas des bâtiments basses consommations [SOLE] Par ailleurs, les données météorologiques disponibles (températures extérieures et flux solaires) sont des données horaires.

On peut ainsi visualiser la variation de la température intérieure des zones et vérifier la stratégie de régulation adoptée (Figure 92). Après une phase transitoire, la température se stabilise autour de la température de consigne (ici 20° avec une tolérance de +1°) grâce à la régulation du chauffage. Par contre, l'état du chauffage n'est mis à jour qu'à chaque pas de simulation, qui correspond dans ce cas à une heure. Pendant ce temps, la température intérieure suit son évolution et peut dépasser les limites tolérées du confort et gêner l'habitant, ou simplement donner des résultats de simulation faux vis-à-vis d'une commande réelle. En effet, cette simulation ne reflète pas suffisamment le comportement réel du système du chauffage dont le changement d'état se fait plus rapidement conduisant à un comportement thermique de la maison, qui dépend de la puissance de chauffage injectée, faux.

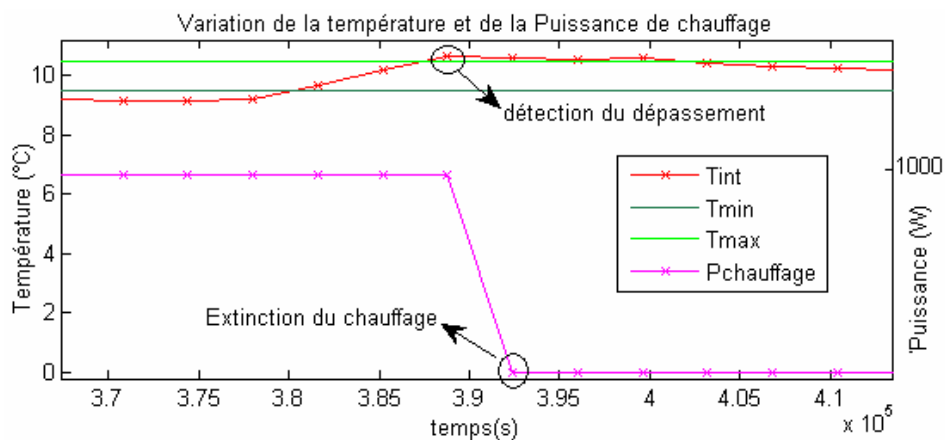


Figure 92. Résultats de la simulation, à pas fixe d'une heure, avec dépassement de seuil géré

II.3.1.2. La simulation à pas fixe de 6 minutes

Afin de remédier à ce problème, le pas de simulation va être réduit à un pas mieux adapté au système de chauffage. On opte ici pour un pas de simulation de 6 minutes qui correspond au pas minimal de simulation dans COMFIE.

Cette réduction du pas de temps va permettre d'améliorer la performance de la simulation rendant ces résultats plus proches de la réalité, lisser les courbes de variations des températures et avoir plus de précision étant donné que les dépassements sont détectés plus tôt avec un pas de temps plus faible (Figure 93).

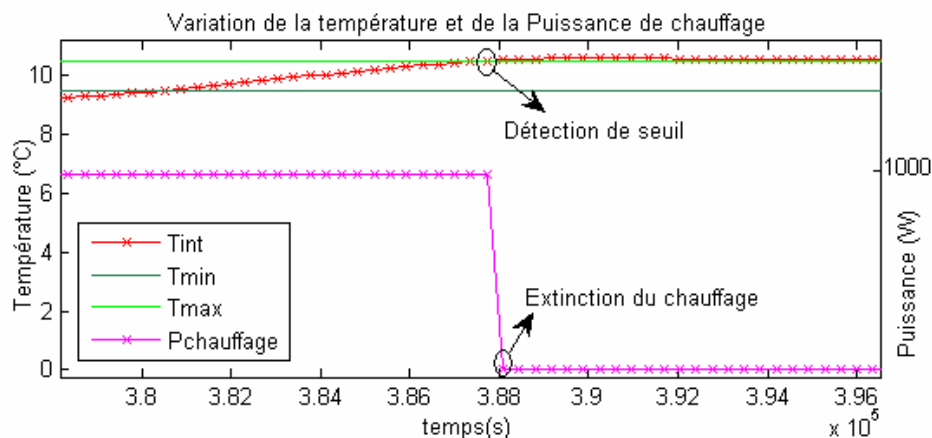


Figure 93. Résultats de la simulation à pas fixe de 6 minutes, permettant une gestion fine de la régulation, mais qui requière un temps de simulation long

En effet, une méthode de simulation à pas fixe offre une précision ayant l'ordre du pas de simulation [PAUL 1999], imposant ainsi une réduction du pas de simulation à la valeur minimale de pas (entre tous les pas du système) si l'on veut obtenir la meilleure précision possible. Il en résulte que l'amélioration des performances (en terme de précision) est pénalisée par une détérioration des temps de calcul : le temps de simulation a été multiplié par dix avec le passage d'un pas d'une heure à 6 minutes. Il faut considérer par ailleurs que cette réduction de pas de calcul, se traduisant par une augmentation du temps de simulation, n'est pas toujours nécessaire, surtout quand aucun événement n'intervient pendant une longue période. Par conséquent, il est important de procéder à une simulation à pas variable permettant d'adapter le pas de temps en fonction de la dynamique du système.

II.3.2. La simulation à pas variable

II.3.2.1. Le principe de la simulation à pas variable

La figure ci après (Figure 94) montre le principe d'une intégration à pas de temps variable. Elle nécessite par exemple le calcul des dérivées des variables d'état, mais aussi, dans le cas d'un système hybride, de méthodes permettant de localiser les événements. Ces méthodes permettent d'alerter le solveur pour qu'il puisse adapter le pas de simulation, en le réduisant à une faible valeur, près de l'instant d'un événement prédit (adaptation par prédiction) ou lors de la détection d'un dépassement de seuil « zero-crossing » (adaptation par correction). Un faible pas de calcul sera adopté à ce moment pour mieux détecter les événements et les changements d'états.

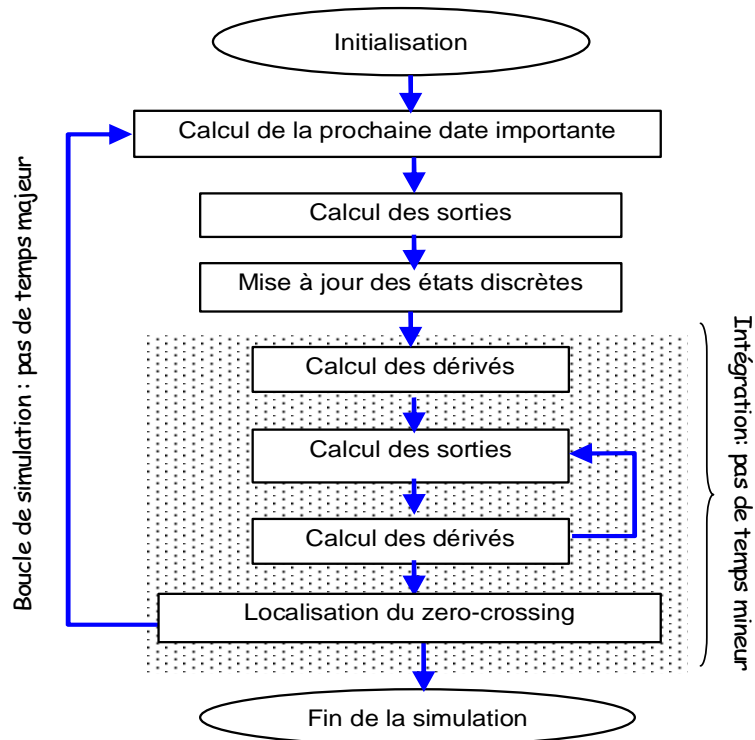


Figure 94. Le principe de la simulation à pas variable dans SIMULINK

De l'autre côté, le modèle doit répondre aux sollicitations du solveur à chaque pas de temps (Figure 95). Il doit donc implémenter des méthodes pour fournir les sorties demandées en fonction des entrées perçues à chaque appel.

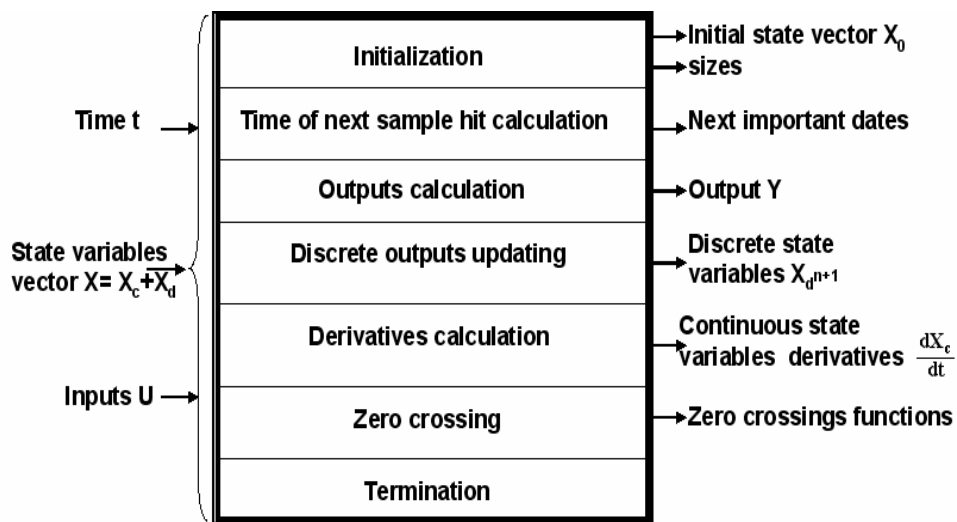


Figure 95. Les ports et les méthodes typiques d'un modèle ODE hybride

II.3.2.2. Les adaptations des natures des modèles nécessaires pour une simulation à pas variable

Afin de procéder à une simulation hybride et pouvoir facilement faire varier le pas de temps, le modèle thermique du bâtiment doit être décrit sous un formalisme ODE (cf chapitre 1, Eq 1). Etant donné que le système étudié précédemment est un modèle discrétisé à un pas fixe, le système différentiel (chapitre 2, Sys 2 et que nous rappelons ci-dessous) va être utilisé

en vue de construire une S-fonction continue à partir de sa représentation d’état. Le solveur de Matlab permettra, par la suite, en fonction de la dynamique du système global (enveloppe + chauffage) de gérer le pas de simulation.

$$\begin{cases} \dot{X} &= F \cdot X + B^\alpha \cdot \dot{U} + B^\beta \cdot \dot{Y} \\ Y &= H \cdot X + S^\alpha \cdot U + S^\beta \cdot Y \end{cases}$$

Or, ce système n'est pas sous la forme d'une représentation d'état classique de la forme $\begin{cases} \dot{X} = A \cdot X + B \cdot U \\ Y = C \cdot X + D \cdot U \end{cases}$ puisqu'il utilise en particulier des dérivées temporelles du vecteur

de sortie Y. Il a donc fallu procéder à un changement de variable $Z = X - B^\beta \cdot Y$ (qui sera notée par la suite X). On aboutit alors au système final suivant :

$$\begin{cases} \dot{Z} &= F \cdot Z + B^\alpha \cdot \dot{U} + F \cdot B^\beta \cdot Y \\ Y &= (Id - S^\beta + H \cdot B^\beta)^{-1} \cdot (H \cdot Z + S^\alpha \cdot U) \end{cases} \quad \text{Sys 4}$$

Dans ce nouveau système subsiste la dérivée du vecteur des sollicitations U, qui sera calculée par la pente entre deux points de calcul consécutifs. Ce terme rend le système sensible aux variations brusques des sollicitations. La température extérieure et les flux solaires incidents ayant des variations faibles entre deux pas de calcul, ils ne perturbent pas remarquablement la résolution, contrairement aux termes relatifs aux puissances internes, liés en particulier à la mise en route du chauffage. En effet, lors d’un dépassement du seuil de température, l’état du chauffage varie ainsi que sa puissance dégagée. La puissance injectée dans le vecteur de sollicitation varie ainsi fortement (passe de 0 W à 1 kW dans ce cas) ce qui provoque des instabilités dans la résolution (Figure 96) [GAA 2010].

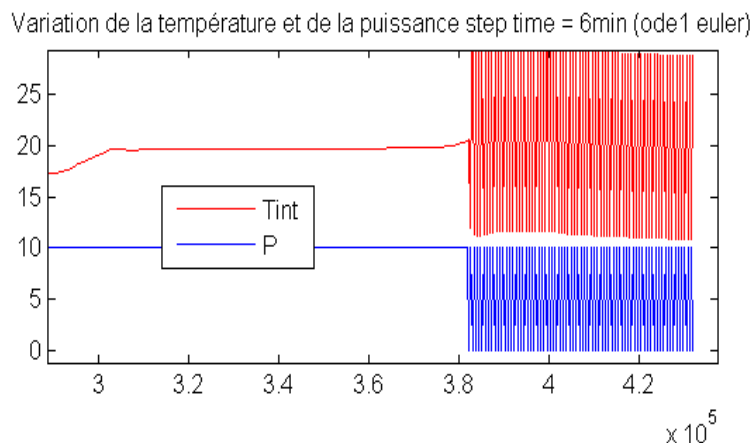


Figure 96. Les oscillations de la puissance et la température dues à une variation brutales des entrées

Afin de stabiliser la simulation numérique, le modèle du chauffage a été affiné par l'introduction de constantes de temps thermiques (Figure 97). Le chauffage se comporte comme un filtre du second ordre réduisant ainsi les variations brutales de puissance lors de la mise en marche et de l’extinction du chauffage. D’un autre côté, ce modèle traduit d’une manière plus proche la réalité du comportement du chauffage étant donné que la puissance

électrique consommée au niveau d'un équipement n'est pas convertie d'une manière instantanée en puissance thermique au niveau de l'enveloppe du bâtiment..

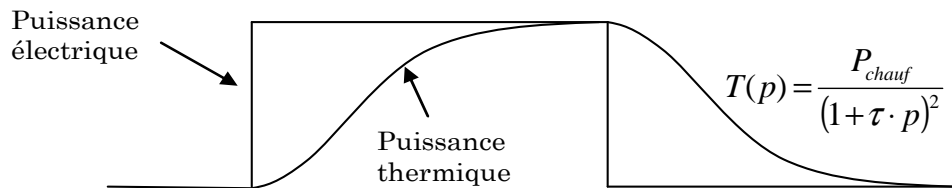


Figure 97. Le modèle de chauffage du deuxième ordre

Il est donc important d'utiliser des modèles avec des niveaux de finesse cohérents, si on veut les connecter afin d'assurer à la fois, une meilleure représentation de la réalité, et aussi une meilleure stabilité numérique.

II.3.2.3. Résultats de la simulation à pas variable

Pour comparer les performances entre une simulation à pas fixe et une simulation à pas variable (entre 6 min et 1 h), il a fallu se mettre dans les mêmes conditions et donc procéder à une nouvelle simulation à pas fixe de 6 min du modèle de l'enveloppe discrétisé couplé au nouveau modèle du chauffage (2^{ème} ordre). Le temps de la simulation avec ce pas fixe est d'environ deux secondes de calcul pour deux semaines simulées.

La figure suivante (Figure 98) montre que le solveur adopte un pas élevé sur la quasi-totalité de la simulation est le réduit au minimum lors de la détection d'une variation dans la dynamique du système et particulièrement lors des commutations du chauffage.

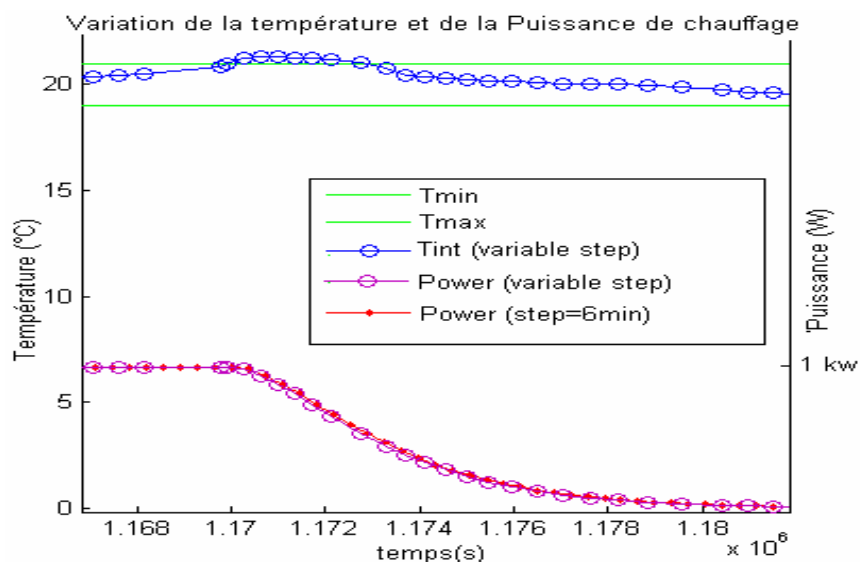


Figure 98. Les résultats de la simulation à pas variable comparées à ceux à pas fixe de 6 minutes

Cette technique a permis l'amélioration des performances de la simulation en réduisant de moitié le temps de calcul tout en assurant une bonne précision dans la détection des événements (par rapport à une simulation à pas de temps d'une heure).

Cette amélioration des temps de calcul n'est pas très significative pour une courte durée de simulation (deux semaines dans ce cas), mais devient plus importante pour une étude s'étalant sur une ou plusieurs années, qui est l'ordre de grandeur des échantillons de simulation essentiellement en phase de design et de prototypage.

Nous allons nous appuyer sur ce cas test, ainsi que sur les travaux précédents du G2ELab [DO, 2010], afin de définir la sémantique des interfaces de la norme ICAr pour les systèmes ODE hybrides. Cette sémantique doit répondre aux besoins de la simulation dynamique et doit être générique de manière à s'adapter à plusieurs types de modèles. L'introduction de l'approche composant dans le processus de simulation permettra de lui garantir plus d'automatisation et de généricité.

Dans la suite, nous allons mettre en œuvre l'approche par composants logiciels qui sera appliquée à des éléments de ce système. Nous allons détailler les interfaces de la facette de simulation dynamique du composant logiciel permettant de supporter des modèles continus et/ou discrets avec détection des événements pour pouvoir assurer leur simulation à pas variable.

II.4. Exploitation de l'approche composant logiciel pour la simulation dynamique hybride

Comme détaillé dans le chapitre précédent, le composant ICAR est un composant multi-facettes. Ainsi, pour répondre au besoin de la simulation dynamique, une facette permettant d'encapsuler des modèles décrits sous le formalisme de description explicite ODE a été définie et implémentée.

Nous allons donc étudier les caractéristiques de cette facette pour montrer sa capacité à accueillir plusieurs types de modèles pouvant être utilisés lors d'une simulation dynamique à pas variable. Cette approche sera donc appliquée, dans un premier temps, au cas d'étude spécifié, pour montrer son intérêt et conclure sur sa capacité à s'étendre facilement à plusieurs autres cas.

II.4.1. La facette de simulation dynamique des modèles ODE de l'ICAR

II.4.1.1. Les ports de la facette :

Un standard de composants logiciels est caractérisé par une standardisation des entrées fournies par l'environnement extérieur ainsi que des sorties exposées pour une utilisation externe. Pour pouvoir différencier automatiquement les différents vecteurs d'entrées/sorties, la norme ICAR propose de les « typer » selon leurs fonctionnalités. Les entrées du composant ICAR sont spécifiées de la manière suivante (Figure 99):

- X : valeurs du vecteur d'états de type «StateVariableArray».
- U : valeurs des sources du modèle de type «Double». Les sources sont différenciées entre elles grâce aux noms qui leur sont attribués.
- P : valeur des paramètres du modèle. A l'état actuel, L'ICAR ne différencie pas les paramètres (constants) et les sources (variables dans le temps) alors que leur séparation s'avère indispensable pour faciliter la projection dans plusieurs outils. On se propose ainsi de les séparer en leur associant un nouveau type « Parameter ».

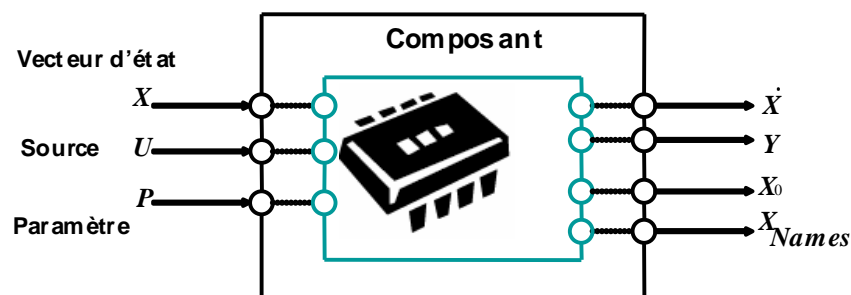


Figure 99. Les ports des entrées et des sorties du composant ICAR pour la simulation dynamique de type ODE

A la sortie du composant on retrouve les sorties suivantes :

- Y : la valeur des sorties du modèle pouvant être observées ou connectées à d’autres modèles.

- \dot{X} : le tableau des dérivées du vecteur d’état qui est également de type « StateVariableArray ».

- X_{names} : le tableau de type « String[] » des noms des variables d’états.

- X_0 : le tableau de valeurs par défaut du vecteur d’état initial, identifié par le type « StateVariableInitialValueArray ».

Ces entrées/sorties peuvent être différenciées en ports de connexion horizontale entre le composant et d’autres modèles (U, Y) et en ports de connexion verticale entre le modèle et un solveur de résolution (X_0, X, \dot{X}) (cf chapitre 3).

II.4.1.2. Les interfaces de la facette

La norme ICAr des composants logiciels pour la simulation des ODE se doit également de spécifier ses interfaces de communication avec un solveur externe (Figure 100). Pour un modèle continu, il s’agit principalement des interfaces d’initialisation en utilisant la méthode « initialize », de calcul des dérivés des variables d’états avec la méthode « computeDerivatives » et le calcul des sorties continues à l’aide de « computeContinuousOutputs » implémentées dans l’interface « ContinuousExplicitStateSystem ».

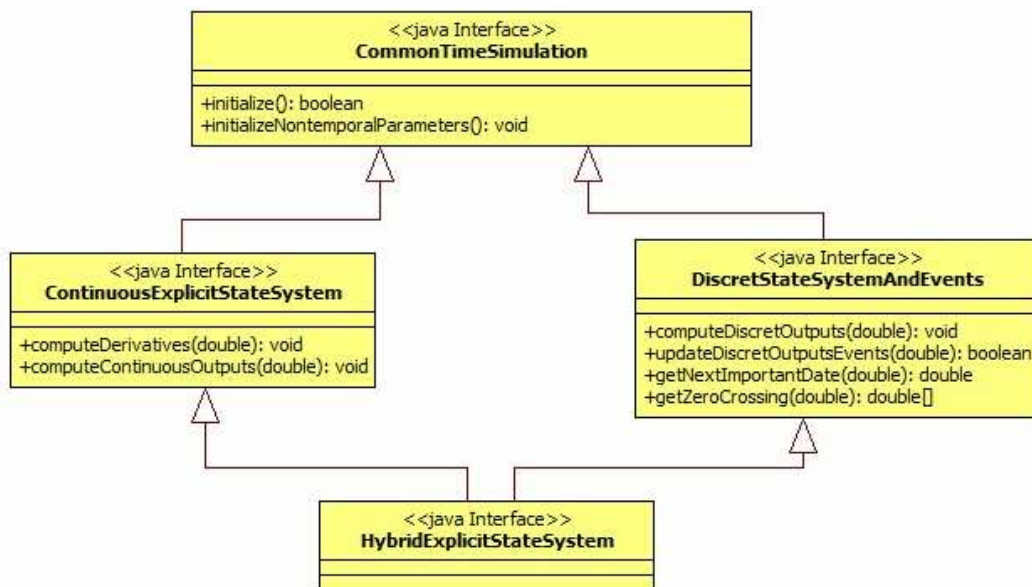


Figure 100. Le diagramme UML de la facette de simulation dynamique ODE de l’ICAr

Lorsqu’il s’agit d’un modèle discret échantillonné ou à événements, d’autres méthodes plus adaptées peuvent être utilisées. En plus de l’initialisation du modèle, qui s’appuie sur la même méthode « initialize » de l’interface commune « CommonTimeSimulation », des méthodes plus spécifiques de l’interface « DiscretStateSystemAndEvents » sont employées comme la mise à jour des variables d’états discrètes grâce à

« updateDiscretStatesAndEvents » ou le calcul des sorties discrètes du système avec « computeDiscretOutputs ». D’autres méthodes sont également implémentées pour alerter des événements qui peuvent avoir lieu et permettre ainsi une simulation à pas adaptatif comme « getNextImportantDate » pour le calcul de la prochaine date importante et « getZeroCrossing » pour déterminer les fonctions à surveiller afin de détecter leurs changements de signe (zero-crossing [WEB_ZER]).

II.4.2. Application de l’approche composant logiciel au cas d’étude

II.4.2.1. La démarche de mise en place

La mise en place d’une simulation, en passant par le concept de standard de composants logiciels, a nécessité la création préalable des plug-outs de génération des composants depuis les modèles utilisés et les plug-ins pour importer ces composants vers les environnements cibles. Pour réaliser le cas test étudié, on s’est appuyé sur le plug-out de COMFIE, ultérieurement développé, pour générer automatiquement le composant logiciel correspondant au modèle de l’enveloppe. Le plug-in de Matlab/Simulink a été également employé pour importer ce composant et le coupler au modèle du chauffage pour pouvoir par la suite le simuler dans cet environnement (Figure 101).

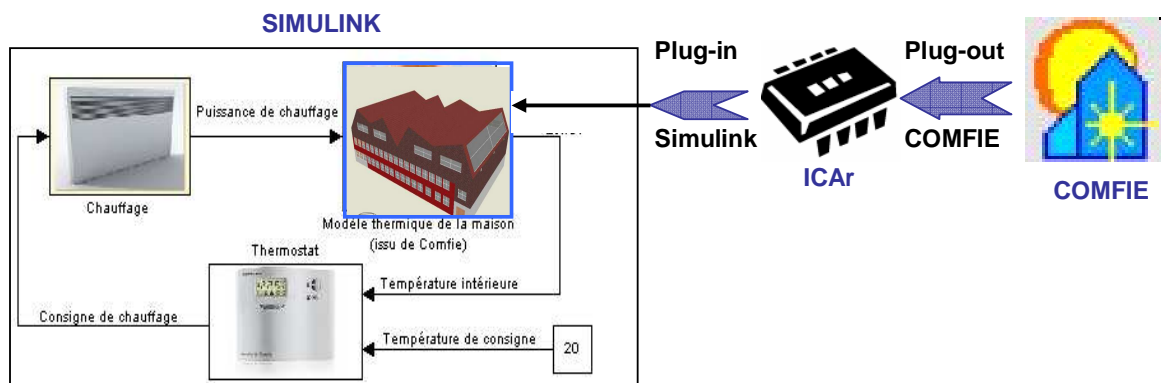


Figure 101. La démarche de réalisation de la simulation du cas test en passant par l’approche ICAR

II.4.2.1.1. Encapsulation du modèle COMFIE dans un ICAR

Cette étape consiste à créer le composant logiciel correspondant au modèle COMFIE grâce au plug-out décrit en annexe (Annexe 9). L’implémentation de la facette ODE et la génération de l’ICAR se fait de manière automatique à partir des fichiers des matrices générées par COMFIE. Il s’agit donc d’un composant modèle (cf chapitre 3) encapsulant le modèle uniquement sans solveur ni algorithme de résolution. Nous allons détailler le processus de cette implémentation pour expliquer l’utilisation des interfaces de la facette ODE de l’ICAR.

Le modèle étudié est un système continu décrit par une représentation d’état ce qui impose l’utilisation de l’interface « ContinuousExplicitStateSystem ». Il suffit ensuite de compléter les différentes méthodes de cette interface avec les instructions nécessaires pour effectuer l’opération désirée. A l’étape d’initialisation, les différentes matrices d’états sont créées à partir des fichiers COMFIE et les vecteurs utilisés sont initialisés. Les méthodes « computeDerivatives » et « computeContinuousOutputs » assurent respectivement le calcul

des dérivées des variables d’états et des sorties à un instant donné (date) à partir de l’équation correspondante du système d’état adapté comme indiqué dans la figure suivante (Figure 102).

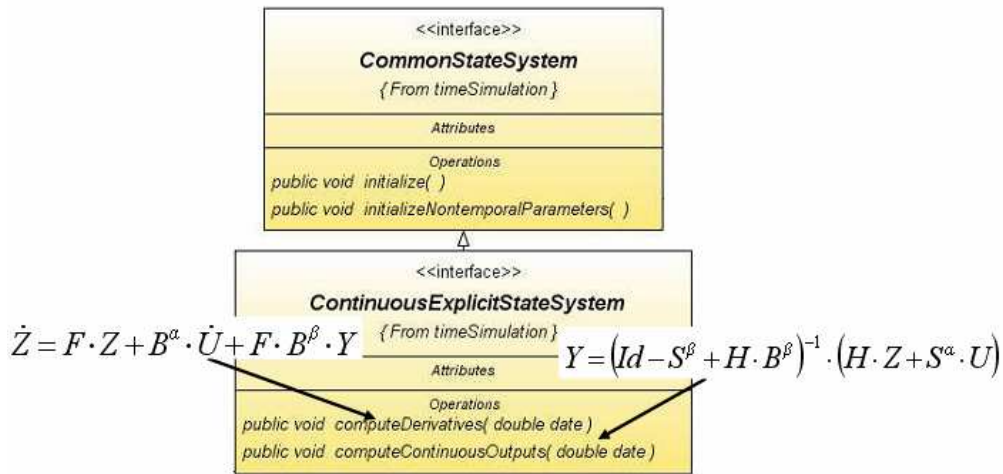


Figure 102. Implémentation de l’interface ODE de l’ICAr COMFIE

II.4.2.1.2. Import de l’ICAr COMFIE dans Simulink

Une fois généré, le composant ICAr, qui est une entité autonome de déploiement, peut être utilisé dans Matlab/Simulink grâce au plug-in développé dans cet outil (Annexe 7). Ce dernier réalise automatiquement la correspondance entre la S-Fonction et l’interface implémentée dans l’ICAr à chaque appel du solveur (Figure 103).

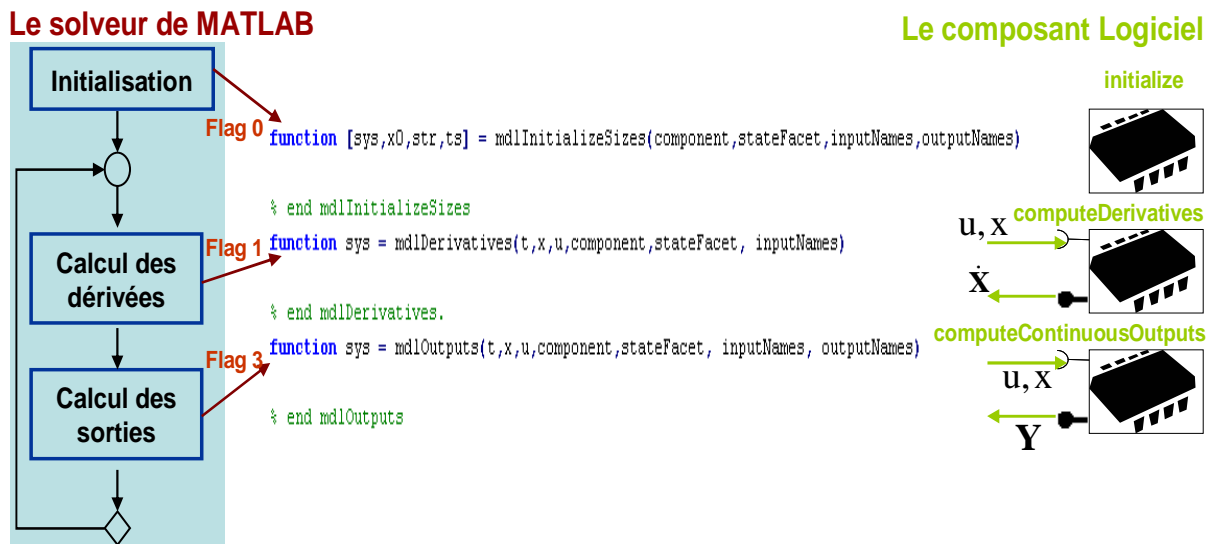


Figure 103. L’interaction entre l’ICAr encapsulé dans une S-fonction et le solveur de Matlab lors d’une simulation ODE continue

II.4.2.2. Les résultats de la simulation

La S-fonction du plug-in, jouant le rôle d’intermédiaire entre le composant ICAr du modèle COMFIE et l’environnement Simulink, est ensuite couplée au modèle de chauffage, en remplacement de la S-fonction, purement matlab, exploitée dans la première partie de ce chapitre.

Les résultats de la simulation de ces deux systèmes sont similaires (Figure 104) avec des faibles différences qui peuvent être dues à des précisions de calculs différentes entre Java et Matlab. Par contre, les temps de calcul ont été doublés avec l'introduction de l'approche ICAR (de 0.16 secondes : modèle sans ICAR à 0.28 secondes : modèle avec ICAR, pour simuler 2 jours). Cette perte est principalement due aux traitements supplémentaires introduits par le passage par la machine virtuelle Java de Matlab pour la communication avec l'ICAR.

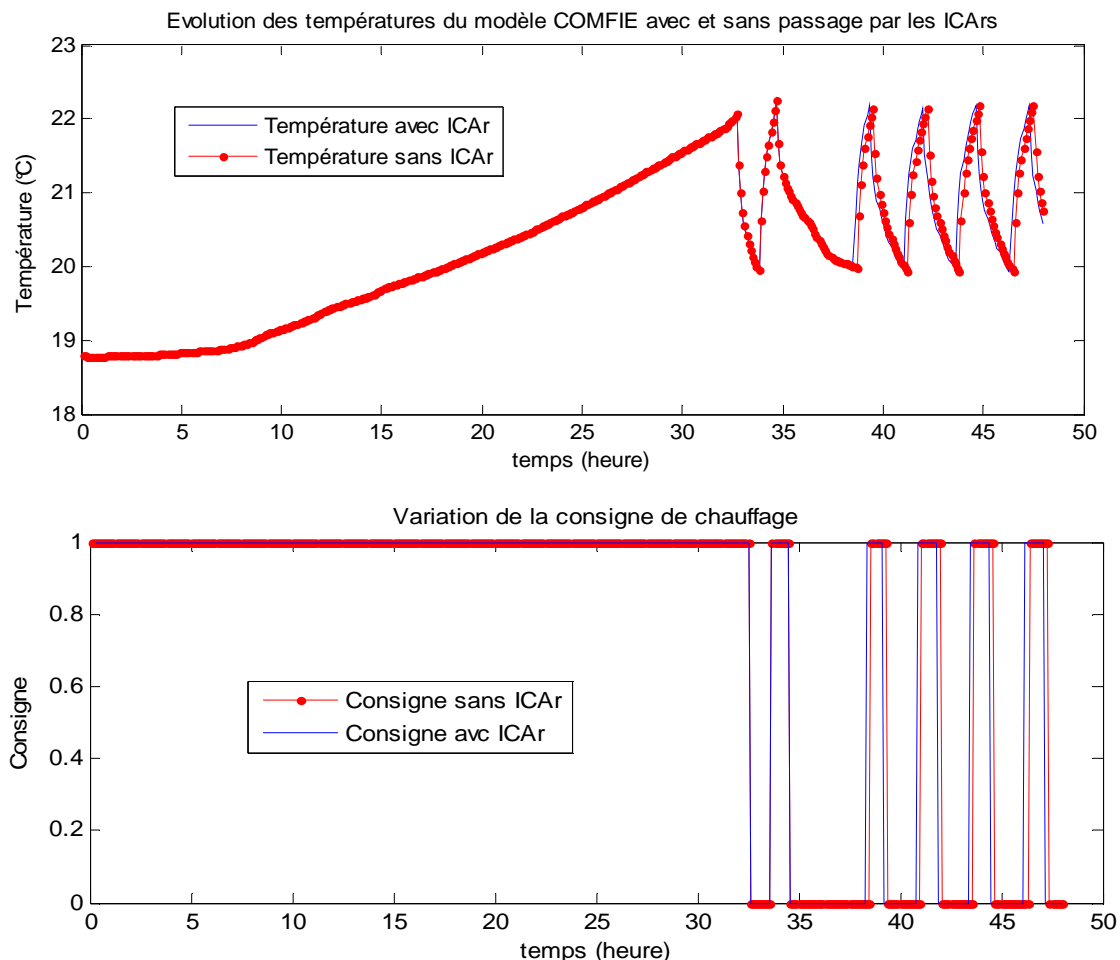


Figure 104. Les résultats de simulation de deux systèmes avec et sans passage par l'approche composants logiciels

II.4.3. Intérêt de l'application de cette approche composant

L'introduction de l'approche composant logiciel se basant sur la norme ICAR dans le processus de simulation dynamique n'influence pas remarquablement les résultats de la simulation (résultats et temps de simulation).

En contre partie, cette procédure assure une mise en place automatique de la simulation grâce au plug-in et au plug-out développés

L'approche composant permet également de garantir une utilisation générique des modèles. En effet, un composant généré peut être utilisé de la même manière dans plusieurs outils grâce aux plug-in qui leurs sont dédiés. Le composant ICAR issu de COMFIE, par

exemple, a été utilisé dans Matlab mais peut être également utilisé dans d'autres outils comme Dymola. Dans les mêmes conditions de simulation, les résultats de simulation du même composant dans des outils différents sont similaires (Figure 105).

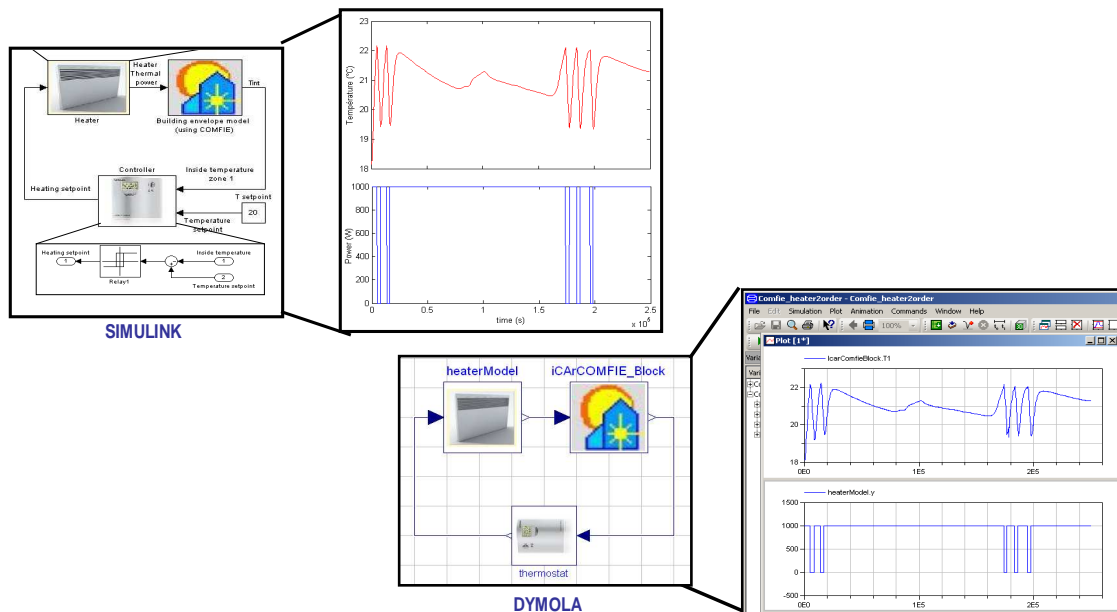


Figure 105. Comparaison des résultats de simulation avec importation de l'ICAr-COMFIE dans Matlab/Simulink et Dymola

De l'autre côté un plug-in développé vers un outil donné est capable d'importer et simuler automatiquement n'importe quel composant (de même type). Le plug-in Matlab, par exemple, a permis d'importer un composant logiciel généré depuis COMFIE et peut aussi importer, de la même manière, un composant issu de BRAHMS. Cet exemple sera détaillé dans la partie suivante.

En outre, les pertes en terme de temps de simulation induites par l'introduction du concept de composant logiciel et observées sur cet exemple simple, peuvent être compensées grâce à une architecture logiciel distribuée. Des ICAr peuvent ainsi tourner en parallèle et à distance sur des web-serveurs et profiter des avantages offerts par cette technologie [DEL 2012].

II.5. Conclusion

Dans cette première partie, les intérêts de la méthode de simulation à pas de temps adaptatif capable de garantir une simulation rapide tout en assurant une précision fine dans la détection des événements, ont été montrés à travers un exemple de système hybride couplant un modèle continu d'enveloppe thermique et un modèle événementiel de chauffage. Cette étude nous a permis d'identifier les besoins de la simulation à pas variable ainsi que la structure d'un modèle hybride en terme de ports et d'interfaces. Cette identification, en complément des travaux déjà réalisés au G2ELab [DO 2010], nous a permis de spécifier la facette de composants logiciels ICAr dédié à la simulation dynamique des ODE hybrides.

L'exploitation de l'approche composant logiciel basée sur la norme ICAr a permis d'assurer une mise en place plus automatisée de ce type de simulation grâce aux plug-ins et plug-outs développés. Elle garantit également un aspect générique car le plug-in Matlab, par exemple, permet d'accepter n'importe quel type de modèle ICAr ODE (continu, discret et hybride) sans besoin d'effort d'adaptation nécessaire.

Plusieurs types de composants peuvent être également générés depuis le plug-out COMFIE : ODE continu et discret et co-simulation. Ces composants peuvent être facilement utilisés, de manière fiable, dans plusieurs environnements de simulation comme Dymola et TRNSys, à travers les plug-ins développés. Ainsi, l'introduction de l'approche composant pour l'encapsulation du modèle COMFIE, a assurée l'amélioration des capacités d'interopérabilité de l'outil par externalisation des modèles qu'il produit.

Dans le cas de l'application étudiée, le modèle de l'enveloppe extrait de COMFIE a été couplé à un modèle de chauffage régulé de manière simple (en hystérésis). D'autres algorithmes de contrôle commande spécifiques et optimisés peuvent être mis en œuvre et testés profitant ainsi des capacités de Malab/Simulink dans ce domaine. Des modèles plus développés, comme le modèle de la VMC double flux peuvent être également couplés au modèle de l'enveloppe du bâtiment, de la même manière, dans Matlab ou dans d'autres outils de simulation.

III. La co-simulation des systèmes hétérogènes

III.1. Introduction

La co-simulation est une simulation co-opérative entre au moins deux, voire plusieurs, solveurs ou outils de simulation (outil : solveur + interface graphique + données) qui tournent simultanément et s'échangent des informations de manière collaborative. Elle peut s'effectuer entre deux outils exécutés sur la même machine physique ou sur deux machines distantes (simulation distribuée) [FUJ 2003].

III.1.1. Les intérêts de la co-simulation

Cette technique de simulation présente plusieurs avantages par rapport à une simulation classique monolithique se basant sur l'intégration et la simulation des modèles issus de différents outils de modélisation dans le même simulateur [FUJ 2005] [TRCK 2006]. Parmi ces avantages on peut citer:

- Un travail collaboratif permettant d'utiliser plusieurs outils à la fois, spécialisés dans des domaines différents, et dépasser ainsi les limitations imposées par l'utilisation d'un unique outil en terme de capacité de modélisation et de simulation.
- Combinaison de plusieurs paradigmes de modélisation (multi-agent, équations différentielles : ODE, DAE...), niveaux de finesse des EDP (mono zone, éléments finis...) et méthodes de résolution (solveurs à pas fixe ou à pas variable, solveurs avec des ordres d'intégration variés...).
- Prototypage facile de nouveaux systèmes grâce aux possibilités accrues d'interopérabilité entre des outils métiers
- La facilité d'entretien et d'amélioration des modèles qui se font généralement depuis leurs propres outils, et qui sont ensuite réintégrés dynamiquement au système via la co-simulation.

III.1.2. Typologie des formes de co-simulation

La co-simulation peut être réalisée de manière directe entre deux outils ou de manière indirecte en passant par un environnement de co-simulation généralement appelé orchestrateur (algorithmes de synchronisation) et le couplages entre plusieurs outils à la fois est possible dans ce cas (Figure 106).

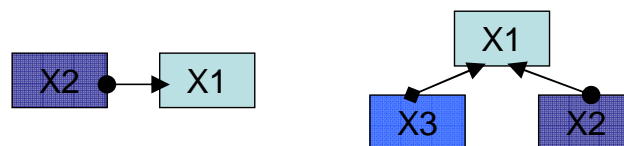


Figure 106. La co-simulation directe (gauche) et la co-simulation indirecte (droite)

Etant donné ces multiples intérêts, la co-simulation a été largement appliquée, à travers le couplage direct entre deux outils ou le développement d'environnements de co-

simulation, dans plusieurs secteurs de l'ingénierie comme l'automobile (Flowmaster²³), l'aéronautique (CosiMate²⁴), la défense (ADvantage²⁵), la mécatronique [GONZ 2010], la chimie [HILL 1986], l'électronique et la microélectronique [RUEL 2002] [LEMAR 2000], les réseaux électriques [MERD 2011], la terre : climat, météo, océan (ESMF²⁶)...

Dans le secteur du bâtiment, les utilisateurs ont eu recours, à plusieurs reprises, à la technique de co-simulation pour coupler les outils nécessaires à la mise en place d'une simulation complète. Ces cas de co-simulations établis dans le bâtiment peuvent être répertoriés selon les deux typologies de co-simulation:

- Une co-simulation directe entre deux outils dans une architecture maître/esclave: comme l'exemple lors de l'intégration du simulateur énergétique du bâtiment « ESP-r » avec le simulateur de lumière « Radiance » [JAN 1999] ou également avec le simulateur dynamique des fluides « Fluent » [DJU 2003]. Le simulateur des systèmes énergétique « TRNSYS » a aussi été couplé avec d'autres programmes comme « EES » [KEI 2002] et « Matlab » [REID 2004].
- Une co-simulation orchestrée par un environnement de co-simulation : l'exploitation de cette solution s'est élargie avec le développement d'un environnement spécifique au domaine du bâtiment, le BCVTB [WET 2008], tenant compte de ses particularités et mettant en place des couplages avec des outils de simulation du bâtiment comme TRNSYS, EnergyPlus, Dymola... Depuis, plusieurs couplages entre plusieurs simulateurs ont été testés pour différents cas d'études [SAG 2011] [PAN 2011] [GU 2011]...

Dans la première catégorie, l'utilisateur se doit de réaliser au cas par cas le couplage entre les deux outils dont il a besoin, ce qui s'avère une tâche fastidieuse demandant des efforts considérables de développement. La deuxième solution présente plus de facilités en terme d'implémentation étant donné que les différents outils de la co-simulation seront reliés au même environnement de co-simulation. L'utilisation d'une approche composant logiciel dédié, standardisant les interfaces des outils, va permettre d'automatiser, de généraliser et de faciliter la mise en œuvre de ces deux solutions.

III.1.3. Intérêts de la co-simulation à base de composants logiciels

Pour faciliter la mise en œuvre de ces co-simulations, plusieurs communautés ont noté le besoin d'une standardisation. Dans le secteur automobile, par exemple, des efforts considérables ont été effectués à travers le projet européen Modelisar [WEB_MODE], conduisant au développement des FMI et permettant aux différents partenaires d'échanger des modèles ou de co-simuler leurs outils d'origine. Un autre standard de composants dédié à la co-simulation uniquement a également émergé dans la communauté hydraulique. Il s'agit du

²³ http://www.flowmaster.com/flowmaster_cosimulation.html

²⁴ <http://www.chiastek.com/products/cosimate.html>

²⁵ http://www.adi.com/products_sim.htm

²⁶ ESMF: Earth System Modeling Framework, <http://www.cisl.ucar.edu/research/2005/esmf.jsp>

standard OpenMI²⁷ se basant sur le langage Microsoft C# et est donc dépendant de cette plateforme d'utilisation. Un autre standard de composant plus générique, le CCA²⁸, a été développé pour des applications nécessitant de grosses capacités de calcul. Il a été par exemple appliqué aux sciences de la terre [LAR 2004].

En effet, un outil de simulation peut être acteur d'une co-simulation s'il peut offrir des interfaces de communication avec son solveur afin de le démarrer et de l'arrêter depuis l'extérieur. Cependant, les interfaces fournies par les différents outils sont très variées au niveau de la syntaxe, de la nature des données, etc. Ceci va nécessiter un traitement et un développement au cas par cas. La mise en œuvre d'une co-simulation dans un orchestrateur X1 (maître) et deux autres modèles X2 et X3 (esclaves) se fait généralement de manière séparée (Figure 107 gauche).

En vue de faciliter et généraliser la procédure de co-simulation, une technique se basant sur le standard de composant logiciel, peut être utilisée. Un standard de composant logiciel pour la co-simulation signifie essentiellement une normalisation des interfaces de communication, qui va rendre leur utilisation dans d'autres environnements plus facile. De cette manière, la co-simulation traitée précédemment peut être développée de la façon suivante : X2 et X3 sont encapsulés sous la forme d'un composant logiciel, donc ils ont les mêmes interfaces. Il suffit alors de développer un unique connecteur du composant logiciel vers X1 qui permettra à la fois de charger et de communiquer avec X2 et X3 (Figure 107 droite).

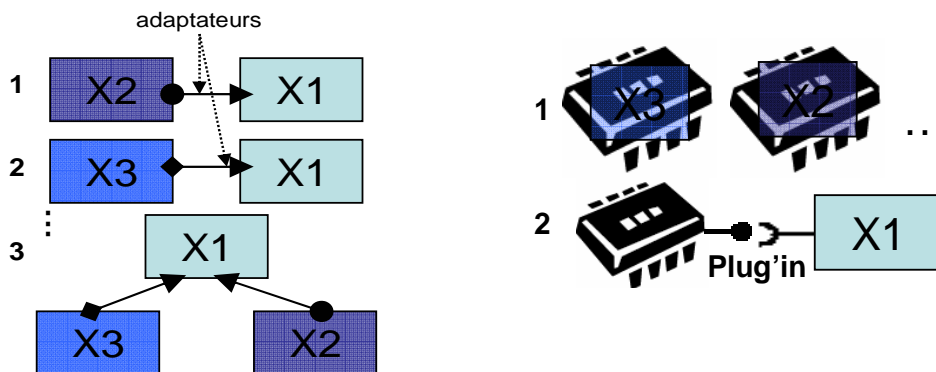


Figure 107. La co-simulation sans et avec l'utilisation d'un standard de composant logiciel.

Une facette dédiée à la co-simulation a été implémentée dans le standard des composants logiciels ICAr afin de répondre à ces besoins. Les détails de cette facette ainsi que le processus de son utilisation et l'intérêt de cette approche seront montrés à travers un exemple de co-simulation réalisé entre COMFIE et BRAHMS dans l'environnement Simulink. Cette étude se focalisera sur la standardisation des ports et des interfaces des acteurs de la co-simulation de manière à pouvoir être utilisés dans le cadre de plusieurs cas de couplage et avec plusieurs stratégies de co-simulation.

²⁷ OpenMI : Open Modeling Interface: www.openmi.org

²⁸ CCA: Common Component Architecture, <http://www.cca-forum.org/overview/index.html>

III.2. Architecture du cas d’étude: Couplage entre COMFIE et Brahms dans Simulink

III.2.1. Description du cas d’étude

Dans ce cas d’étude, on vise à simuler le même bâtiment que l’exemple traité dans la partie précédente avec le même niveau de finesse mais en tenant compte de l’occupant qui peut intervenir dynamiquement sur la consigne de chauffage quand elle dépasse son seuil de confort. Donc le modèle de l’enveloppe sera construit à partir des matrices d’états de COMFIE en utilisant le système discret (chapitre 2, Sys 3) rappelé ici (avec un pas d’échantillonnage d’une heure) permettant de définir un modèle de co-simulation (il est résolu récursivement sans besoin d’un solveur d’intégration externe).

$$\begin{cases} X^{n+1} = \exp(F \cdot \Delta t) \cdot X^n + W^\alpha \cdot (U^{n+1} - U^n) + W^\beta \cdot (Y^{n+1} - Y^n) \\ Y^{n+1} = MGIF \cdot X^n + MGIE \cdot U^{n+1} + MGID \cdot U^n - MGIC \cdot Y^n \end{cases}$$

Le modèle de chauffage du deuxième ordre est conservé mais il ne sera plus contrôlé par un thermostat, c’est l’occupant qui agira directement sur sa consigne.

Cet usager est modélisé dans Brahms par des règles de déplacement entre les deux zones du bâtiment et l’extérieur et aussi par des actions sur la consigne de chauffage de la zone dans laquelle il est [GAA 2012]. Sa présence dans une zone, induit l’injection d’une puissance supplémentaire dans le vecteur des sources qui correspond à la chaleur que dégage son corps. Les interactions entre les différents modèles sont représentées dans le schéma suivant (Figure 108) :

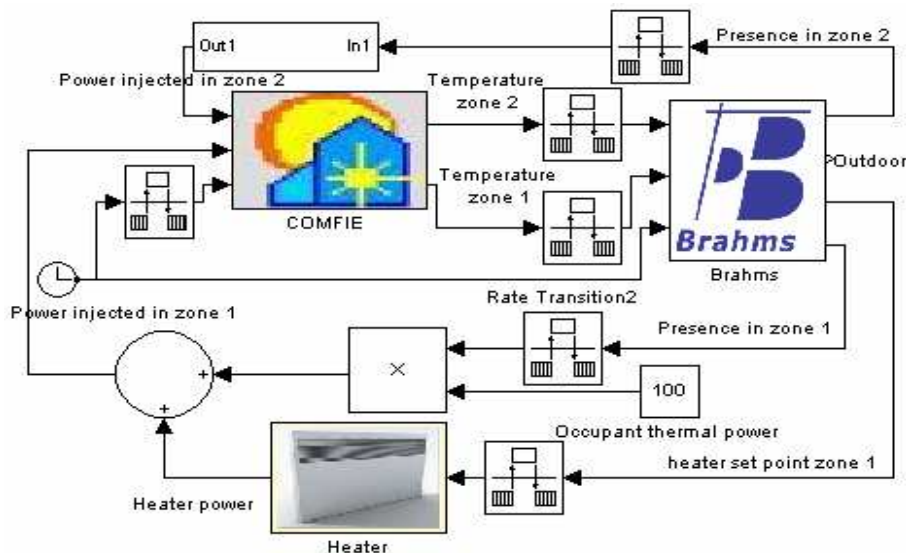


Figure 108. Le schéma de la co-simulation entre COMFIE et BRAHMS orchestrée dans SIMULINK.

Il s’agit donc d’un modèle d’enveloppe issu de COMFIE couplé à un modèle d’occupant simulé dans Brahms et un modèle de chauffage modélisé dans Matlab/Simulink. L’ensemble est orchestré par l’environnement Simulink (co-simulation indirecte).

III.2.2. les intérêts de ce couplage

COMFIE permet une modélisation assez fine de l'enveloppe thermique du bâti. Comme plusieurs autres logiciels de la simulation énergétique du bâtiment, il ne néglige pas l'utilisateur étant donné qu'il prend en compte la puissance thermique qu'il dégage à travers des scénarios prédéfinis avant la simulation. Par contre, cette modélisation ne prend pas en considération la complexité de la variation dynamique du comportement de l'utilisateur. Ce dernier point est en effet majeur par rapport :

- aux sources d'incertitudes induites par le comportement de l'utilisateur dans la simulation du bâtiment et l'estimation de l'influence de ceci sur la consommation globale

- à la modélisation et à la prise en compte du confort [CAN 2005].

En effet, l'approche de modélisation par scénario ne permet pas de représenter l'interaction de l'utilisateur avec son environnement et ne permet donc pas, entre autre, de vérifier le respect de son confort. Ceci constitue un paramètre sensible nécessitant une étude dès la phase de conception.

Or, l'amélioration des outils pour la prise en compte de l'utilisateur va nécessiter des efforts considérables au niveau méthodologique ainsi qu'au niveau de l'implémentation, alors qu'il existe d'autres outils spécialisés dans ce domaine qui pourraient les compléter pour dépasser leurs limites, comme le logiciel Brahms.

Les travaux de Kashif et Le [KASH 2011] [LE 2010] ont démontré la capacité de l'outil BRAHMS à assurer une simulation performante du comportement de l'utilisateur dans les bâtiments pour une meilleure gestion de l'énergie. BRAHMS est une plateforme multi-agent permettant de simuler le comportement réactif et délibératif d'un occupant. L'occupant y est modélisé comme un agent qui est conscient de son environnement et peut donc réagir sur ses objets (ici le chauffage) et modifier leurs états en fonction des informations perçues (la température) comme nous l'avons décrit dans le chapitre 2.

Contrairement à Comfie qui offre la possibilité d'en extraire les capacités de modélisation, Brahms n'autorise pas une séparation entre le modèle et son solveur (la machine virtuelle de Brahms) pour une résolution externe. Par conséquent, les modèles de Brahms ne peuvent être simulés que dans leur propre environnement. La co-simulation s'avère donc l'unique solution permettant d'avoir une simulation détaillée à la fois de l'enveloppe du bâtiment via COMFIE et du comportement des usagers à l'aide de BRAHMS.

La difficulté de cette co-simulation réside dans un couplage de deux modèles de nature différente (discret/événementiels, pas de simulation, paradigme de modélisation, etc.). COMFIE est un simulateur physique étudiant des phénomènes physiques alors que Brahms est un simulateur multi-agent basé sur des règles de comportement.

Le modèle issu de COMFIE est un modèle discret (échantillonné à pas d'une heure) et les modèles de Brahms sont événementiels (actions des agents) résolus en temps discret. Les événements de Brahms ne sont mis à jour qu'à chaque pas de temps (ici 60 secondes: pas de l'horloge). L'orchestration de ces modèles doit donc assurer une bonne gestion du pas de

synchronisation de manière à détecter, de manière précise, les événements de Brahms. Trois stratégies d'orchestration seront proposées par la suite.

Ces problèmes d'interopérabilité seront traités en application à ce couplage, grâce au concept de composant logiciel. Ce concept sera ensuite généralisé pour le rendre accessible à plusieurs applications et outils de simulation.

III.3. Orchestration des acteurs de la co-simulation : exemples de stratégies d'orchestration

Dans ce paragraphe, on propose quelques stratégies possibles d'orchestration des deux modèles autonomes qui seront étudiées dans le but de déterminer les fonctionnalités qui doivent être implémentées dans un composant de co-simulation pour qu'ils soient compatibles avec diverses stratégies de synchronisation et pour définir par la suite les interfaces de la facette de co-simulation du composant ICAR.

III.3.1. La synchronisation à pas fixe

Une stratégie basique peut être tout d'abord évoquée quand il s'agit de deux simulateurs ayant le même pas de temps et pour lesquels la synchronisation entre eux se fait à chaque pas du temps (ou à un multiple du pas de temps). Or le pas de temps du modèle COMFIE est fixé à une heure, ce qui n'est pas adapté à une simulation de Brahms et ne permet pas de bien détecter les événements qu'il génère.

Dans un premier temps, on suppose que la synchronisation entre les deux outils n'est assurée qu'au pas de temps le plus grand qui est celui du modèle COMFIE (Figure 109). Avec cette stratégie, l'utilisation d'un pas de temps réduit pour la simulation de Brahms n'a donc pas d'influence sur les résultats globaux de la simulation et ne permet pas une amélioration de la précision vu que la synchronisation entre Brahms et COMFIE ne se fait que toutes les heures même si un événement de Brahms est détecté entre temps.

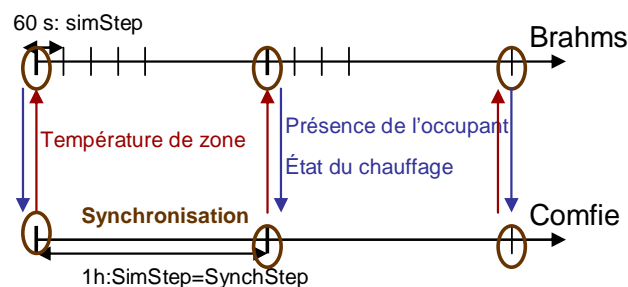


Figure 109. La synchronisation à pas fixe entre COMFIE et BRAHMS.

Cette co-simulation couplant deux simulateurs de constante de temps différente est une co-simulation de type «Multi-rate» ; multi-cadences. Ce type de co-simulation est souvent rencontré dans les systèmes complexes couplant des composants issus de disciplines différentes et donc caractérisés par des variations temporelles différentes [SOL 2009]. Plusieurs méthodes de couplages traitant ces différences ont été étudiées depuis les années 70 [GEAR 1984] [GONZ 2010] [ELL 2000] [GRAV 2001] [TRCK 2008] [BOU 2006] se basant principalement sur l'interpolation et l'extrapolation des données.

III.3.2. La synchronisation à pas variable

On peut classer ces différentes stratégies selon deux principales catégories. Des stratégies de synchronisation par prédiction se basant généralement sur des méthodes d'extrapolation pour prédire le prochain instant de synchronisation avec la possibilité de réduire le pas de simulation autour de ce point. Ces stratégies de co-simulation ne sont pas étudiées ici et ne seront pas supportées par la norme actuelle de l'ICAR de co-simulation.

On trouve aussi des stratégies de co-simulation par correction se basant généralement sur des méthodes de retour en arrière. Certaines méthodes ont été appliquées dans notre cas d'étude. La faisabilité de ces méthodes a été vérifiée, cependant leurs mises en œuvre en vue d'une comparaison de leurs performances, ne sont pas actuellement réalisées car l'utilisation de Simulink ne le permettait pas facilement, ce qui aurait donc nécessité le développement d'algorithmes de synchronisation spécifiques plus adaptés à ces stratégies.

III.3.2.1. La synchronisation avec un retour en arrière

Cette stratégie se base sur le principe du retour en arrière au pas de synchronisation qui précède la détection d'un événement tel qu'un franchissement de seuil des niveaux des températures. L'orchestrateur peut interrompre, quand c'est possible, la simulation dès la détection d'un événement (interruption de la simulation de Brahms sur un déplacement entre les zones par exemple) ou autoriser les acteurs à finir leurs calculs et aboutir au prochain pas de synchronisation. Lorsqu'un événement est détecté, un retour en arrière s'impose pour pouvoir redémarrer à partir de l'instant d'occurrence de l'événement. .

Plusieurs solutions sont envisageables pour assurer un bon « re-démarrage », en fonction des possibilités offertes par le modèle. Une première solution serait que COMFIE reparte du dernier instant de synchronisation en adoptant un pas de simulation et de synchronisation plus faible pour mieux détecter l'événement (Figure 110). Une deuxième solution consiste simplement en une interpolation permettant de détecter l'instant du franchissement de seuil et de relancer ainsi les modèles à cet instant avec le même pas de temps.

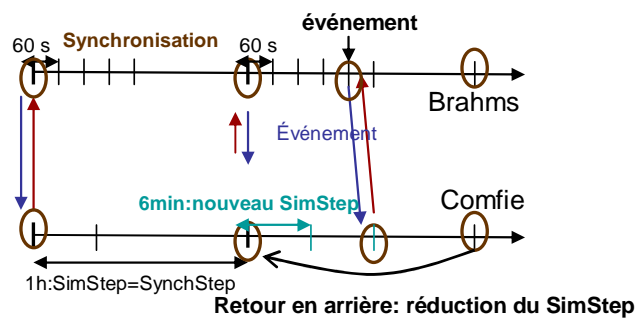


Figure 110. La synchronisation par retour en arrière lors de la détection d'un événement

III.4. Exploitation de l'approche composant logiciel pour la co-simulation

III.4.1. La facette de co-simulation de l'ICAr

La définition de la facette de co-simulation « CosimulationSlave » du composant ICAr a nécessité l'étude de plusieurs APIs offertes par plusieurs solveurs, ainsi que plusieurs stratégies de couplages possibles, comme nous l'avons détaillé dans le paragraphe précédent: avec un pas de synchronisation fixe ou variable avec interruption du processus ou avec retour en arrière... Ceci est dans le but d'avoir une norme générique.

Comme chaque facette de la norme ICAr, la facette de co-simulation est définie par des ports de connexion et des interfaces de communication (Figure 111).

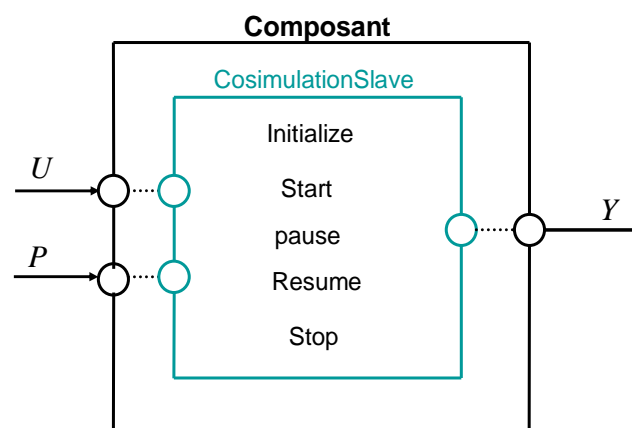


Figure 111. L'interface de co-simulation de l'ICAr.

A l'entrée du composant on retrouve :

- U , le vecteur des sources du modèle
- P : les paramètres du modèle

A la sortie du composant on a définit:

- Y , le vecteur des sorties.

Les principales interfaces implémentées dans la facette de co-simulation sont:

- Initialize (): permet d'initialiser le modèle (esclave) et préparer les éléments dont il a besoin pendant la simulation (bibliothèques, fichiers, machine virtuelle, configuration...)
- Start (starTime, simStep, synchTime, InitialValues): Elle permet de lancer ou de relancer (en cas de retour arrière) une simulation à partir d'une date starTime, avec un pas de simulation simStep jusqu'à l'instant de synchronisation synchTime en lui affectant les valeurs initiales dont il a besoin. Cette méthode est indispensable dans le cas du retour en arrière par exemple où le modèle repart d'un instant non prévu et nécessite de prendre en compte les entrées (sources) à cet instant précis.

- Resume (simStep, synchTime): relance la suite de la simulation du modèle à partir de l'instant où elle s'est normalement arrêtée (sans interruption ni retour arrière) jusqu'à l'instant de synchronisation synchTime et avec un pas de simulation simStep
- Pause (): Permet d'interrompre, quand c'est possible, le processus de simulation avant sa fin.
- Stop (): permet de terminer la simulation en assurant par exemple la libération des mémoires qui ont pu être allouées ou en supprimant des fichiers qui ne sont plus nécessaires...

Toutes ces méthodes permettant de renseigner leurs états de déroulement à travers une variable booléenne « StepIsDone ». Cette variable contrôlée par l'orchestrateur lui permet d'autoriser le passage à l'étape suivante tout au long du processus de la simulation et d'appel des différentes interfaces.

Le composant contient également plusieurs méta-données dans le but de renseigner l'utilisateur sur le contenu du modèle (qui peut être une boîte noire pour lui). Ces informations vont permettre une meilleure connaissance des capacités du composant pour mieux gérer la connexion du composant et son orchestration à travers des stratégies avancées. Parmi les méta-informations on peut citer:

- la capacité de varier son pas de temps ou pas
 - o le pas de temps du modèle (si pas de temps fixe),
 - o les pas de temps supportés par le modèle ou son intervalle de variation (si pas de temps variable),
- la possibilité d'interrompre une simulation,
- la capacité de se réinitialiser...

III.4.2. Application de l'approche composant logiciel au cas d'étude

Dans cette partie, la stratégie de synchronisation à pas fixe (Figure 109) sera adoptée. Le pas de simulation de Brahms a été fixé à 60 secondes pour qu'il puisse avoir un échange régulier avec Matlab afin de l'informer des variations du comportement de l'occupant. Le pas de simulation de COMFIE est fixé par le modèle à 1 heure. Cette différence au niveau des pas de simulation est gérée par Simulink via les blocks « rate Transition » [WEB_RAT] qui permettent l'adaptation des données échangées entre les deux acteurs.

III.4.2.1. La démarche de mise en place

Afin de réaliser le couplage décrit précédemment dans l'environnement Simulink et en passant par l'approche composant, plusieurs étapes seront suivies. La première étape consiste à la génération des composants logiciels demandés depuis leurs outils d'origine grâce aux plug-outs dédiés.

Ensuite, les deux composants seront importés en tant que « S-fonctions » dans Matlab/Simulink via le même plug-in générique (Figure 112). Ils peuvent ainsi être connectés au modèle de chauffage et simulés dans cet environnement.

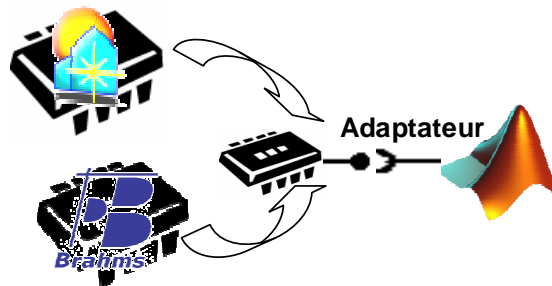


Figure 112. Principe de la procédure de mise en place du couplage

III.4.2.1.1. Le plug-out de COMFIE

Pour construire le composant de co-simulation de COMFIE, le plug-out automatique depuis cet outil (Annexe 9) est utilisé. Le plug-out COMFIE va procéder à l'extraction des matrices générées et la reconstruction en Java du système d'état discret (chapitre 2, III.1.2 : [3]). Ce système est discrétisé, il n'a donc pas besoin d'un solveur de résolution pour s'exécuter. Il peut être ainsi considéré comme un composant autonome (composant-simulateur).

Contrairement à la facette de simulation dynamique ODE, aucune adaptation sémantique n'est nécessaire pour définir les interfaces du composant. Il suffit, ainsi, de compléter les interfaces nécessaires du composant avec les informations convenables à partir des équations du système d'état.

Il s'agit essentiellement de définir les matrices d'états à partir des fichiers des matrices à la méthode *initialize()*, calculer la sortie *Y* et mettre à jour le système d'état à chaque appel de la méthode *resume()* (ou *start()* pour la première itération). Le pas de simulation est imposé par le modèle et renseigné en tant que méta donné.

III.4.2.1.2. Le plug-out de Brahms

Ce plug-out automatique permet la génération du composant intégrant à la fois le modèle de l'occupant ainsi que sa machine virtuelle de résolution : composant-simulateur (cf chapitre 3).

Il assure une adaptation syntaxique entre les ports de l'ICAr et les interfaces offertes par Brahms pour la communication avec ses modèles en lecture et en écriture (*exemple* : *setAgentAttributeDouble*, *getAgentLocation*, *getObjectAttributeString...*) pour définir ses différentes entrées/sorties.

Cette adaptation est aussi assurée entre les interfaces de l'ICAr et les interfaces de commande externe la machine virtuelle de Brahms, à chaque étape de la simulation. A l'initialisation du composant, la machine virtuelle de Brahms (BVM) est lancée en imposant la date de son démarrage. La simulation peut être ensuite démarrée, arrêtée ou reprise grâce aux API suivantes : *JAPI.getVMController().start()*, *JAPI.getVMController().resume()*,

JAPI.getVMController().stop()...Des informations sur la fin d'une étape de calcul (StepIsDone) ou sur la détection d'un événement sont également récupérées...

La synchronisation entre Brahms et l'extérieur peut se faire en utilisant l'API « *AwaitTimeAdvance* » qui surveille la simulation et notifie tous les événements. On peut ainsi arrêter la BVM dès que l'instant de l'évènement détecté est supérieur ou égal au temps de synchronisation désiré (*synchTime*).

III.4.2.1.3. Le plug-in Simulink

De la même manière que pour le plug-in de la facette ODE, un plug-in pour la facette de co-simulation est réalisé. La S-fonction discrète (aucune variable continue) (Annexe 7) est employée à la place de la S-fonction continue pour permettre au composant d'interagir avec le solveur. Il s'agit principalement d'exécuter *initialize()* et *start()* au premier appel du solveur (flag 0) et ensuite *resume()* à chaque demande de calcul des sorties (flag 3) tout en affectant les nouvelles entrées au composant et récupérant les dernières sorties.

Une fois réalisé, ce plug-in permet automatiquement : d'importer, d'introspecter et d'utiliser plusieurs composants logiciels de la même manière, ce qui peut nettement faciliter la mise en œuvre de plusieurs cas de co-simulation. Ainsi, les ICAR de COMFIE et de Brahms peuvent être utilisés à l'aide de ce même plug-in et simulés dans Simulink en tant que des S-fonctions classiques (Figure 112).

III.4.2.2. Résultats de la co-simulation

Les S-fonctions correspondantes aux ICAR de COMFIE et Brahms peuvent être connectées entre elles et à d'autres composants de la bibliothèque Simulink de manière à aboutir au schéma global du système désiré (Figure 108). Les principaux résultats de cette co-simulation sont présentés dans la figure ci-dessous (Figure 113).

Dans la première courbe, on peut observer la variation de la présence de l'occupant dans la zone principale (zone 1). On y retrouve les scénarios des déplacements prédéfinis dans le modèle de BRAHMS (cf chapitre 2). On observe également sur la troisième courbe, la variation de la consigne de chauffage donnée par l'occupant selon le dépassement des limites de température représentées sur la deuxième courbe: Extinction si $T > T_{max} = 20.5^{\circ}\text{C}$ et mise en marche si $T < T_{min} = 19.5^{\circ}\text{C}$. Par contre, si l'occupant n'est pas dans la zone chauffée, il ne peut plus agir sur la consigne même si la température dépasse le seuil de confort et elle continue donc d'augmenter pour atteindre des niveaux plus hauts, comme ce qui s'est passé à l'instant $t=14\text{h}$.

On peut par exemple envisager de comparer l'impact énergétique de deux comportements différents de l'occupant celui qui est représenté ici et un autre, plus conscient énergétiquement, qui éteint le chauffage avant de quitter le local pour une longue durée.

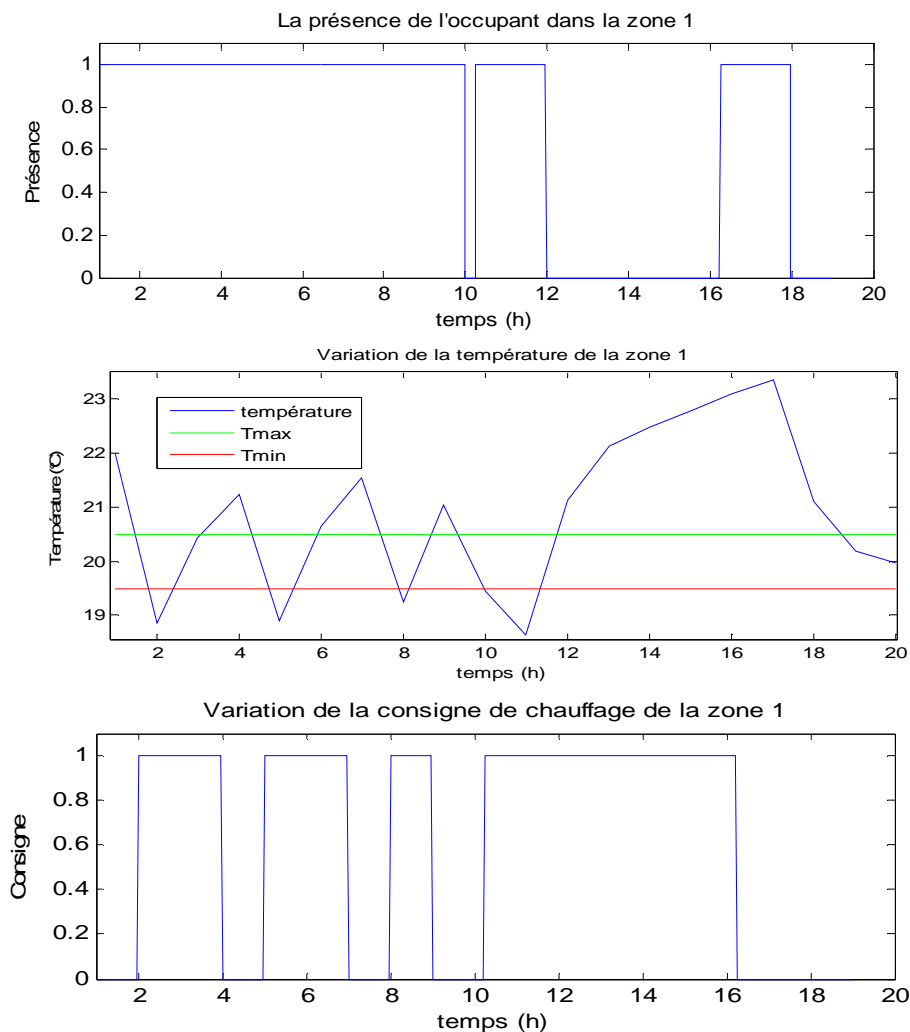


Figure 113. Les résultats de la co-simulation entre COMFIE et Brahms dans Simulink

La simulation de 20 heures du système complet a duré environ 20 minutes et donc la simulation d'une année complète nécessite des journées entières. Cette lenteur de la simulation globale est due principalement à la synchronisation entre Simulink et Brahms (essentiellement l'arrêt et la reprise de la BVM), et ne remet pas en cause l'approche par composants logiciels. La simulation d'un modèle de Brahms depuis Simulink a multiplié par 10 les temps de calcul par rapport à une simulation lancée directement de Brahms, qui était déjà assez lente (2 minutes pour une journée). Le temps de simulation du modèle de COMFIE dans Simulink reste assez faible en comparaison à une simulation dans son propre environnement: quelques dixièmes de secondes pour simuler une journée entière. Afin d'améliorer les performances de cette co-simulation, l'outil Brahms (open source) devrait être retravaillé pour optimiser les séquences successives de reprise de simulation.

III.4.3. Les intérêts de la co-simulation par l'approche composant

Bien que les temps de simulation aient été affectés, l'introduction de l'approche composant logiciel dans le processus de co-simulation a montré plusieurs avantages :

- une architecture générique et dynamique de co-simulation : cette approche a permis de coupler automatiquement, en utilisant le même plug-in vers l'environnement Simulink, deux composants logiciels issus de deux outils différents.
- Une gestion de l'hétérogénéité des simulateurs : elle a permis, facilement, le couplage entre deux formalismes différents : la simulation physiques dynamique, ayant vocation à simuler des phénomènes physiques ; et la simulation par agents, ayant vocation à simuler des systèmes collaboratifs notamment des acteurs humains.
- Complémentarité des outils pour dépasser leurs limites mutuelles : La co-simulation a assuré la complémentarité entre ces deux outils (COMFIE et Brahms) pour profiter à la fois des capacités d'une modélisation fine du bâti et d'une modélisation du comportement de l'occupant prenant compte de son environnement, de son confort et de sa réaction.
- Une possibilité de parallélisation à travers les web-services. Cette perspective offre la possibilité d'utilisation de ressources (des logiciels, des ordinateurs...) partageables et à distance qui permettra la réduction des temps de co-simulation de modèles plus complexes.

III.5. Conclusion

Dans cette dernière partie du chapitre nous avons détaillé plusieurs stratégies possibles de co-simulation à travers l'exemple des outils Comfie et Brahms dans l'environnement Matlab/Simulink. Cet exemple nous a permis de montrer l'intérêt et les fonctionnalités des interfaces développées dans la facette de co-simulation du standard ICAr de composants logiciels.

Cette facette vise à être générique pour pouvoir encapsuler plusieurs types de simulateurs (physiques, multi-agents...) et être adaptée à plusieurs stratégies d'orchestration et de synchronisation possibles. En plus des autres aspects montrés dans la partie précédente (possibilités d'utilisation du composant dans plusieurs environnements, facilité et d'automatisation d'usage), la capacité d'un plug-in, une fois mis en place, à importer plusieurs composants a été mis en évidence. Cette caractéristique représente un avantage important de notre approche pour la mise en place rapide de co-simulations, dans les phases d'esquisse en conception ou réhabilitation par exemple.

Dans l'exemple étudié, un modèle de l'occupant assez simple a été utilisé mais d'autres scénarios de modélisation plus complexes sont envisageables. L'habitant peut être représenté au sein d'un groupe de personnes (typiquement une famille) avec lesquelles il peut délibérer et confirmer ou annuler son action sur un objet. Les objets peuvent être des chauffages mais aussi des portes ou des fenêtres sur lesquels une action peut modifier la structure du modèle du bâtiment, ce qui renforce encore plus la problématique d'interopérabilité.

Ce travail, s'est focalisé essentiellement sur la standardisation des acteurs de co-simulation (les esclaves) sans trop rentrer dans les détails des orchestrateurs et leurs algorithmes qui peuvent être également standardisés pour unifier les structures et les procédures d'appel des acteurs. Une autre perspective peut consister à l'implémentation de plusieurs stratégies de co-simulation comme celles décrites précédemment pour comparer leurs performances respectives.

IV. Conclusion

Deux dimensions scientifiques ont été abordés dans les deux parties de ce chapitre relatives à la gestion et l'orchestration des modèles hétérogènes : l'aspect sémantique ainsi que l'aspect technique et les concepts informatiques de réalisation associés. Ces deux dimensions ont été étudiées à travers deux approches de simulation: la simulation à pas variable des systèmes hybrides et la co-simulation des systèmes hétérogènes multi-cadences.

Au niveau sémantique, l'architecture des ports de connexions et la syntaxe des interfaces de communication ont été identifiées pour chaque approche de simulation, à travers deux cas d'études, afin de répondre aux plus larges besoins d'orchestration. Cette étape d'identification a permis la spécification des facettes de la norme de composant ICAr spécifiques à chaque approche.

Au niveau technique, l'architecture logicielle relative au concept de composant a été appliquée au cas d'étude spécifié pour conclure sur les avantages d'utilisation de cette approche : automatisation et généralisation. Ce chapitre vient donc illustrer les concepts détaillés dans le chapitre 3, montrer leurs intérêts et se focaliser plus particulièrement sur la mise en œuvre des plug-ins et plug-outs au travers des composants logiciels ICAr. Deux types de composants logiciels ont été utilisés : le composant-modèle (dans la simulation hybride à pas variable de COMFIE) et le composant simulateur (co-simulation COMFIE/Brahms).

Le composant-simulateur dédié à la co-simulation présente l'avantage d'une facilité de mise en œuvre car les outils sont traités en externe sans rentrer dans les détails d'hétérogénéité de leurs modèles et de formalismes de description. Le composant-modèle, par contre, doit tenir compte des sémantiques variables des modèles selon leurs natures.

Le composant-simulateur peut également bénéficier de stratégies d'orchestration dédiées et sur mesure, qui sont moins difficiles à mettre en œuvre que des solveurs spécifiques pour simuler les composants-modèles.

Le composant-modèle présente l'avantage d'être plus léger: il comporte les équations nécessaires du modèle, et est donc moins influent sur les temps de simulation en comparaison à un composant-simulateur. Ce dernier est connecté non seulement au modèle mais aussi à son solveur et parfois à l'interface graphique de l'outil, nécessitant donc un nombre important de ressources nécessaires à son exécution.

Dans tous les cas, une solution n'est pas meilleure que l'autre, tout dépend des objectifs visés et des contraintes rencontrées, propres à chaque projet de simulation.

Conclusions et perspectives

Conclusions

Résumé de la problématique traitée

Les travaux présentés dans le présent manuscrit mettent l'accent sur une problématique incontournable liée à la modélisation et la simulation des systèmes de bâtiment. Il s'agit de la problématique d'interopérabilité devenue prioritaire dans un nouveau contexte nécessitant une gestion énergétique globale de tous les systèmes. Pour être efficace, cette gestion passe par une modélisation globale du système « bâtiment » prenant en compte ses différents acteurs, leurs spécificités et leurs interactions.

Or, cette simulation se trouve confrontée à des modèles difficiles à mettre en œuvre, très hétérogènes car issus de différents domaines physiques et des simulateurs très variés et spécialisés dans des domaines précis. Ces simulateurs présentent donc des limites pour assurer une modélisation de tous les acteurs du système.

Résumé des solutions apportées

Deux principales solutions d'interopérabilité ont été exploitées dans ces travaux.

-La première s'appuie sur le langage émergent et multi-physique Modelica. Elle consiste au re-développement (manuel ou automatisé) de tous les modèles nécessaires à la modélisation globale du système dans ce format unique.

-Une deuxième approche a été proposée en complément de la première approche pour dépasser certaines de ces limites. Il s'agit de l'approche se basant sur un standard de composant logiciel. On s'est spécialement intéressé à la norme ICAr développée initialement pour des applications électriques et qui a montré son aptitude à s'étendre vers d'autres domaines et à devenir un standard potentiel d'interopérabilité dans la filiale du bâtiment.

Cette approche boîte noire est capable de compléter la première approche boîte blanche grâce aux notions de « plug-in » de « plug-out » et de bus à composants. Il s'agit d'outils informatiques permettant d'améliorer les capacités des outils existants par un travail collaboratif s'appuyant sur l'échange de modèles.

Les bénéfices des solutions apportées

Les intérêts de l'approche boîte blanche « Modelica » ont été mis en avant à travers la modélisation des composants des systèmes du bâtiment PREDIS. Ce langage a montré sa capacité de modélisation multi-physique de composants issus de différents domaines et de

différentes natures temporelles. Son aspect orienté objet favorise la réutilisation des modèles existants (développés en Modelica).

D'un autre côté, le concept de standard de composant, présente diverses caractéristiques permettant de le placer comme un candidat potentiel pour atteindre les objectifs fixés en terme de modélisation et de simulation système grâce à :

- Un formalisme abstrait lui permettant d'accueillir plusieurs modèles hétérogènes.
- Une approche multi-facette permettant l'encapsulation de plusieurs types de modèles. Cet aspect garantit une adaptabilité du standard ICAr le rendant extensible.
- Des interfaces standardisées et identifiées à partir des modèles existants qui peuvent être automatiquement connectables dans plusieurs outils et environnements.
- Une autonomie permettant aux composants d'être librement échangés directement ou via des sites internet.

Cette solution d'interopérabilité devient plus pertinente avec le développement d'un bus à composant riche en « plug-in » et « plug-out ».

Les limites de ces approches

En contre partie, diverses limites théoriques, techniques et pratiques ont été détectées avec l'utilisation du langage Modelica. Ces limites consistent en certaines difficultés d'usage, de capitalisation et de réutilisation de certains types de composants.

L'approche de composant logiciel, explorée récemment dans la filière bâtiments, présente également certains points à améliorer au niveau de ces plug-ins et plug-outs. Ces points à améliorer concernent la phase de développement (programme de génération automatique de certains plug-ins, sélection automatique des bibliothèques DLL..), la phase de déploiement où beaucoup de travail reste à faire pour mieux automatiser cette étape et la phase d'utilisation...

Perspectives autour de l'approche composant logiciel

Vers une plateforme d'interopérabilité

Les perspectives à court terme consistent à enrichir le bus à composant logiciels construit autour de l'ICAr par divers développements en vue de mieux inciter le déploiement de ce standard dans le secteur du bâtiment. L'environnement de co-simulation BCVTB représente une cible principale de ces futurs travaux, pour profiter des divers développements qui y sont réalisés.

Ce bus est mis en place pour jouer le rôle d'intermédiaire entre les outils de modélisation, de simulation et les langages de description. L'élargissement de ce bus pour couvrir plusieurs outils conduit au développement d'une sorte d'environnement d'interopérabilité capable de favoriser le travail collaboratif entre ces divers outils.

Vers les réseaux collaboratifs

Le standard de composant logiciel ICAr est développé en Java afin d'être compatible avec les nouvelles technologies du réseau internet. Deux pistes sont visées et en cours d'exploration à ce niveau : les web services et les sites Internet collaboratifs comme Dimocode. Les web services permettent de rendre le composant disponible à distance et sur étagère pour être utilisé pour des applications locales. Les sites internet collaboratifs, vont permettre de facilement partager non seulement les modèles mais aussi les connaissances relatives à ces modèles et les avis des experts.

Vers un nouveau standard multi-métier

Une extension de cette norme ICAr de composant logiciel est étudiée en perspective de ces travaux, dans le cadre du projet ANR PLUMES. Il s'agit d'une norme plus générique et multi-métier nommé MUSE (Figure 114). L'objectif est de pouvoir couvrir, en plus des domaines physiques et diverses natures de modèle, plusieurs métiers relatifs au secteur du bâtiment.

Le composant MUSE s'appuie sur l'expérience acquise des composants ICAr, mais adressera 3 métiers spécifiques, à la fois: la simulation dynamique, le dimensionnement optimal, et la gestion optimale.

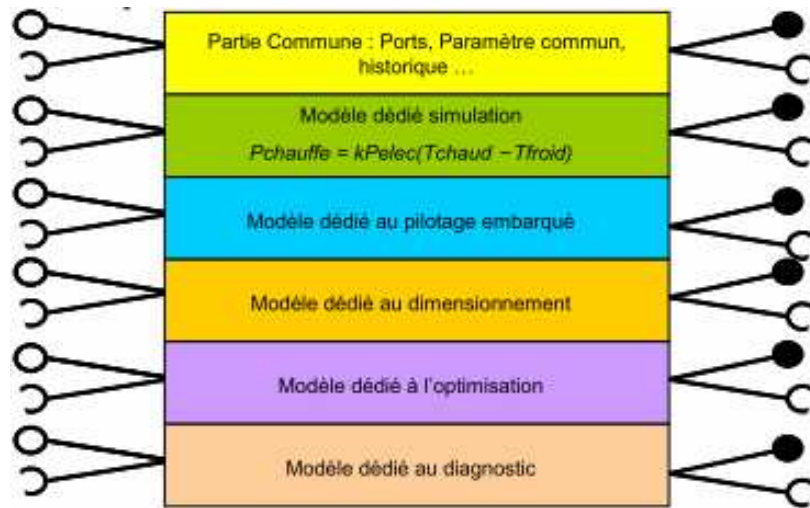


Figure 114. Le composant Muse

Bibliographie

A

- [ABR 2008] S. Abras, S. (2008), «Mutli-Agents Home Automation System for power management in buildings», Thèse de doctorat INPG, 2008.
- [ADEME 2005] Agence De l'Environnement et de la Maîtrise de l'Energie, rapport sur les «Stratégies utilisation rationnelle de l'énergie», Juin 2005.
- [ADEME 2008] centre des ressources pour le climat et l'énergie territoriaux, <http://www.pcet-ademe.fr/domaines-actions/batiments/contexte-et-enjeux>.
- [ADEME 2009] Agence De l'Environnement et de la Maîtrise de l'Energie, «Energie et Climat», chiffres clés, édition 2009.
- [ADEME 2010] Agence De l'Environnement et de la Maîtrise de l'Energie, «Bâtiment Energie Environnement», collection chiffres clés, édition 2010, disponible sur : http://www.cercad.fr/IMG/pdf/ademe-batiment_chiffres-cle_2010.pdf.
- [ADEME 2011] Agence De l'Environnement et de la Maîtrise de l'Energie, «Les chiffres clés du bâtiment», édition 2011.
- [AKES 2008] J. Åkesson, T. Ekman, G. Hedin, «Development of a Modelica Compiler using JastAdd», Electronic Notes in Theoretical Computer Science (ENTCS), Avril 2008.
- [AKES 2009] J. Åkesson, T. Ekman, G. Hedin, «Implementation of a Modelica compiler using JastAdd attribute grammars». Science of Computer Programming, Juillet 2009.
- [ALG] Izuba Energies, «Méthode COMFIE : Cahier d'algorithmes», documentation accessible dans le répertoire d'installation de COMFIE.
- [ALL 2003] L. Allain, «Capitalisation et traitement des modèles pour la conception en génie électrique», thèse de doctorat INPG, 2003.
- [AMB 2011] F. Amblard, N. Ferrand, D. R. C. Hill, «How a conceptual framework can help to design models following decreasing abstraction», conference SCS-European Simulation Symposium, Marseille, Octobre 2001.
- [AMO 2009] EDF (Electricité De France) et AITIF (Association des Ingénieurs Territoriaux de France) «Guide de Conception des bâtiments à Basse consommation d'énergie», http://www.amoes.com/data/document/guide_conception-batiment-bbc_amoies_aitif_edf.pdf.
- [ANN 2006] M. Annabi, A. Mokhtari, T.A. Hafrad, «Estimation des performances énergétiques du bâtiment dans le contexte maghrébin», Revue des Energies Renouvelables, vol. 9, n°2, pp 99-106, 2006.
- [ATT 2010] S. Attia, «Building Performance Simulation Tools: Selection Criteria and User Survey», rapport de recherché, Université catholique de Louvain, Belgique, 2010.
- [ATT 2011] S. Attia, «State of the Art of Existing Early Design Simulation Tools for Net Zero Energy Buildings: A Comparison of Ten Tools, technical report», rapport technique, 2011.
- [AUG 1994] G.L.M. Augenbroe, «An overview of the COMBINE project», Proceedings of the First European Conference on Product and Process Modeling in the Building Industry (ECPPM '94), Germany, pp.547-554, 1994.

[AUG 2002] G. Augenbroe, «Trends in buildings simulation», Building and environment Journal, N° 37, pp 891-902, 2002.

B

[BAC 1984] P. Bacot, «Analyse modale des systèmes thermiques», Thèse de doctorat, Université Paris VI, 1984.

[BARB 2002] F. Barbier et al, «Composants dans l'ingénierie des systèmes d'information : concepts clés et techniques de réutilisation», actes des deuxièmes assises nationales du GdR I3 (Information - Interaction - Intelligence). Nancy, déc. 2002.

[BAZ 1999] V. Bazjanac, D. Crawley, «Industry foundation classes and interoperable commercial software in support of design energy efficient buildings», Building Simulation Conference, Japan, 1999.

[BEN 2010] S. Benslama, «Mise en place d'un système domotique et d'un SCADA pour la gestion des flux énergétiques de la plateforme Monitoring et Habitat Intelligent de Predis», rapport de stage, Polytech' Grenoble, Juillet 2010.

[BER 2001] Koen Bertels and Magnus Boman, «Agent-Based Social Simulation in Markets», Electronic Commerce Research, Volume 1, Numéro 1-2, pp 149-158, 2001.

[BES 2000] P. Bessemoulin, J. Oliviéri, «Le rayonnement solaire et sa composante ultraviolette», La Météorologie 8e série n° 31, disponible sur <http://www.smf.asso.fr/Ressources/Bessemoulinbis31.pdf>, Septembre 2000.

[BOED 2009] D. Boeda, «Etude de la contribution du pilotage de charges à la fourniture de services aux réseaux électriques», Thèse de doctorat INPG, 2009.

[BOU 2006] F. Bouchhima, M. Brière, G. Nicolescu, M. Abid, E. M. Aboulhamid, «A SystemC/Simulink Co-Simulation Framework for Continuous/Discrete-Events Simulation», proceeding of the IEE international Behavioral Modeling and Simulation Workshop, San Jose, Californie, Septembre 2006.

[BOUR 2005] D. Bourgeois, «Detailed Occupancy Prediction occupancy-sensing control and advanced behavioural modelling within whole building energy simulation», thèse de doctorat de la Faculté des études supérieures de l'Université Laval, Québec, Juin 2005.

[BRO 2006] D. Broman and P. Fritzson, «Type Safety of Equation-Based Object-Oriented Modelling Languages», Conference on Programming Language Design and Implementation, PLDI, Ottawa, Canada, 2006.

[BRO 2007] D. Broman, «Safety, Security, and Semantic Aspects of Equation-Based Object-Oriented Languages and Environments», these de doctorat de l'université de linkoping, Suède, Decembre 2007.

C

[CAIRE 2004] R. Caire, «Gestion de la production décentralisée dans les réseaux de distribution», Thèse de doctorat INPG, 2004.

[CAN 2005] R.Cantin, B. Moujalled, Dr HDR G. Guarracino, «Complexité du confort thermique dans les bâtiments », 6ème congrès Européen de Science des Systèmes, Paris- France, Septembre 2005.

[CAV 2011] D. Cavellera, «Contribution au renvoi de tension et à la reconstitution du réseau, Identification des paramètres d'un réseau, Estimation des flux rémanents dans un transformateur», Thèse de doctorat de l'INPG, spécialité génie électrique, Novembre 2011.

[CELL 1991] F.E. Cellier, «Continuous System Modeling», Springer-Verlag New York, chapitre 6, 1991.

[CEL 1996] F. E. Cellier, «Object-Oriented Modeling of Physical Systems: Promises and Expectations», Symposium on Modelling Analysis and Simulation, CESA'96, IMACS Multi-Conference on Computational Engineering in Systems Applications, pp11261127, France, 1996.

[CGDD 2010] Commissariat Général du Développement Durable au ministère de l'écologie, de l'énergie, du développement durable et de la mer, «Bilan énergétique de la France pour 2010», 2010, disponible sur http://www.developpement-durable.gouv.fr/IMG/pdf/Ref_energie_2010.pdf.

[CHAB 1997] B. Chabot, «La nécessité de changer de comportement. Un monde d'énergies à conquérir», Écodécision n° 36, 1997, pp. 11-15.

[CHEN 2010] H. Chenailler et al «Etude pour quantifier la part des apports internes dans bâtiment tertiaire BBC. Application au bâtiment de PREDIS», conférence IBPSA-France, Moret-sur-Loing, Novembre 2010.

[CHEN 2012] H.Chenailler, «L'efficacité d'usage énergétique : pour une meilleure gestion de l'énergie électrique intégrant les occupants dans les bâtiments», Thèse de doctorat de l'INPG, spécialité génie électrique, Avril 2012.

[CITEPA, 2010] Centre Interprofessionnel Technique d'Etudes de la Pollution Atmosphérique, Rapport d'activités 2010, disponible sur: <http://www.citepa.org/publications/rapport%202010v34-1.pdf>.

[CONC 2002] P. Concannon, «Residential ventilation», AIVC Technical Note 57, 2002.

[COU 2002] J. L. Coulomb, «Electromagnétisme et problèmes couplés», Chapitre 8 Optimisation, EGEM Hermes 2002.

[CRAW 2005] D.B. Crawley, J.W. Hand, M. Kummert, B.T. Griffith, «Contrasting the capabilities of building energy performance simulation programs», building simulation conference, Montreal, Canada, Aout 2005.

D

[DANG 2010] H. A. Dang, «Modélisation électrique d'une VMC double flux et optimisation de sa commande pour l'efficacité énergétique», rapport de mastère de l'INPG, spécialité génie électrique, Juin 2010.

[DANG 2012] H. A. Dang, «Gestion de l'énergie électrique dans l'habitat, cas du stockage électrochimique», Journées AUGC-IBPSA, Chambéry, Juin 2012.

[DAUT 1997] S. Dautin, «Réduction de modèles thermiques de bâtiments : amélioration des techniques par modélisation des sollicitations météorologiques», thèse de doctorat, Université de Poitiers, 1997.

[DEEU 20011] statistiques du département de l'énergie des états unis, disponible sur http://apps1.eere.energy.gov/buildings/tools_directory/, 2011.

- [DEL 2003] B. Delinchant, «Un Environnement à base de Composants, Intégrant le Concepteur et ses Outils, pour de Nouvelles Méthodes de CAO», Thèse de doctorat de l'INP Grenoble, spécialité génie électrique, Juillet 2003.
- [DEL 2004] J. Delvare, S. MERCIER Sylvain, «Dimensionnement d'un drone» Rapport de projet de l'ENSIEG (Ecole Nationale Supérieure d'Ingénieur Electriciens de Grenoble) 2004.
- [DEL 2004] B. Delinchant, F. Wurtz, D. Magot, L. Gerbaud, «A component-based framework for the composition of simulation software modeling electrical systems», Simulation, Transactions of the Society for Modeling and Simulation International, Special Issue: Component-Based Modelling and Simulation, June 2004.
- [DEL 2011] B. Delinchant, «La CAO et l'optimisation de systèmes, une approche par couplages dynamiques de composants», rapport de HDR (Habilitation pour Diriger des recherches), spécialité génie électrique, Université de Grenoble, Decembre 2011.
- [DEL 2012] B. Delinchant, S. Gaaloul, et al, «The distribution of computing services for buildings design and optimal management: cloud computing based on software component standard», IECON 2012, Montreal, Canada, Octobre 2012 (non publié).
- [DEL 2012b] B. Delinchant, L. Estrabaud, L. Gerbaud, F. Wurtz F, «Multi-criteria design and optimization tools» chapter 5 of Integrated Design by Optimization of Electrical Energy Systems, Edited by Xavier Roboam, Wiley ISTE, June 2012.
- [DEL 2012c] B. Delinchant, «La CAO et l'optimisation de systèmes, une approche par couplages dynamiques de composants», Habilitation à diriger des recherches, spécialité génie électrique, 8 décembre 2011.
- [DJU 2003] E. Djunaedy, J.L.M. Hensen, M.G.L.C. Loomans, «Towards external coupling of building energy and air flow modeling programs», ASHRAE Transactions, volume 109, numéro 2, 2003.
- [DO 2010] T.P. Do, «Simulation dynamique des actionneurs et capteurs électromagnétiques par réseaux de réductances : modèles, méthodes et outils», thèse de doctorat de l'INPG, spécialité génie électrique, Mars 2010.
- [DUP 2006] B. du Peloux de Saint Romain, «Modélisation des actionneurs électromagnétiques par réseaux de réductances. Création d'un outil métier dédié au pré dimensionnement par optimisation», Thèse de l'Université Joseph Fourier, 2006.
- [DUPR 2006] M. Dupret, «Confort d'été : comment réduire les apports internes dans les bâtiments de bureaux», rapport Enertech, Juin 2006.
- [DYM 2010] Manuel d'utilisation de Dymola, «Dymola User Manual, Volume 2», Janvier 2010.
- [DYMO 2010] Dymola, «Functional Mock-up Interface (FMI) for Model Exchange», manuel d'utilisation des FMI dans Dymola, Fevrier 2010.
- [DYNA 2010] P. Tittlein, P. Riederer, M. Albaric, E. P. Del Bario, «Rapport scientifique final», Projet ANR Dynasimul, disponible sur http://dynasimul.univ-lr.fr/index.php?option=com_docman&Itemid=73, 2010.

E

[EDD 2004] Encyclopédie du développement durable, 2004, <http://encyclopedie-dd.org/encyclopedie/sciences-et-techniques/a-3-faits-et-chiffres/les-consommations-d-energie-dans.html>.

[EFF 2008] EFFINERGIE, «Réussir un projet de Bâtiment Basse Consommation», guide à destination des professionnels du bâtiment, 2008.

[EJJ 2010] K. Ejjabraoui, «Contribution à la conception de systèmes mécatroniques automobiles : méthodologie de pré-dimensionnement multi-niveau multi-physique de convertisseurs statiques», Thèse de doctorat, UPS, 2010.

[ELH 2006] M.S. El Mmam, «Contribution à la modélisation et à la simulation hybride du flux de trafic», Doctorat de l'Université d'Artois, Spécialité : Génie Informatique et Automatique, 2006.

[ELL 2000] A. S. Elliott, «A highly efficient, general-purpose approach for co-simulation with ADAMS», Proceedings of 15th European ADAMS User's Conference, Rome, Italie, consultable sur : http://www.mscsoftware.com/support/library/conf/adams/euro/2000/MDI_Cosimulation.pdf, 2000.

[ELM 1995] H Elmqvist, M. Otter, F. E. Cellier, «Inline Integration: A New Mixed Symbolic/Numeric Approach for Solving Differential–Algebraic Equation Systems». In Proceedings European Simulation Multiconference, Prague, Czech Republic, 1995.

[ELM 1997] H Elmqvist, S. Erik. Mattsson, «Modelica-The next generation modelling language an international design effort», First World Congress on System Simulation (WCSS), Singapore, Septembre 1997.

[ENG 2007] O. Enge-Rosenblatt, J. Haase, C. Clauß, «Important Characteristics of VHDL-AMS and Modelica with Respect to Model Exchange », first International Workshop on Equation-Based Object-Oriented Languages and Tools, Berlin, Allemagne, Juillet 2007.

F

[FEL 2002] F. Felgner, S. Agustina, B. Cladera, R. Mertz, L. Litz, «Simulation of thermal building behaviour in Modelica», la deuxième Modelica conference, Allemagne, 2002.

[FIS 2004] V. Fisher, «Composants logiciels pour le dimensionnement en génie électrique. Application a la résolution d'équations différentielles», Thèse de doctorat de l'INP Grenoble, 2004.

[FMI 2010a] La spécification des fmi pour l'échange des modèles version 1.0, projet Modelisar, http://www.modelisar.com/specifications/FMI_for_ModelExchange_v1.0.pdf, Juillet 2010.

[FMI 2010b] La spécification des fmi pour la cosimulation version 1.0, projet Modelisar, http://www.modelisar.com/specifications/FMI_for_CoSimulation_v1.0.pdf, Juillet 2010.

[FOG 2009] A. Foggia, «Pilotage Optimal de Systeme Multi-sources pour le Batiment », thèse de doctorat de l'INP Grenoble, spécialité génie électrique, Juillet 2009.

[FRIT 2001] P. Fritzson «Introduction to Modelica the Language for Physical Modeling and Simulation», chapitre 1 du livre Modelica, disponible sur <http://www.ida.liu.se/labs/pelab/modelica/modelicaBookChap1-sept01.pdf>, Septembre 2001.

[FRIT 2004], P. Fritzson, «Principles of Object-Oriented Modeling and Simulation with Modelica 2.1», édition 2004.

[FRIT 2006] P. Fritzson, A. pop, P. Aronsson, «OpenModelica Compiler (OMC) Overview», cours des développeurs d'OpenModelica, disponible sur: <http://www.ida.liu.se/labs/pelab/modelica/OpenModelica/OpenModelicaDevelopersCourse/resources/slides/4-OMC%20Compiler-overview-slides.pdf>», 2006.

[FRIT 2006b]:P. Fritzson, «Introduction to Object-Oriented Modeling and Simulation with OpenModelica», tutorial, disponible sur <http://www.ida.liu.se/labs/pelab/modelica/OpenModelica/Documents/ModelicaTutorialFritzson.pdf>, 2006.

[FRIT 2009] P. Fritzson et al, «OpenModelica System Documentation», documentation de OpenModelica version 1.5 Beta, September 2009.

[FUJ 2003] R.M. Fujimoto, «Distributed simulation systems», Proceedings of 2003 Winter simulation conference, The Institute of Electrical and Electronics Engineers (IEEE), New Orleans, Louisiana, USA, 2003.

[FUR 2007] S. Furic, «Hybrid acausal modeling using Modelica». Journées «outils» Lyon: INSA, 2007.

[FUR 2009] S. Furic, «Enforcing model composability in Modelica», 7 ème Modelica conference, Como, Italie, Septembre 2009.

G

[GAA 2010] S. Gaaloul , B. Delinchant, F. Wurtz, S. Thiers et B. Peuportier, «Couplage d'un modèle thermique issu de COMFIE avec un chauffage régulé électriquement pour une simulation temporelle hybride à pas variable», IBPSA_France, Moret sur loing, France, Novembre 2010.

[GAA 2011] S. Gaaloul, B. Delinchant, F. Wurtz, S. Thiers et B. Peuportier, «Strong coupling of a building thermal model with a controlled electrical heater for an hybrid energy simulation», ELECTRIMACS, Cergy Pontoise, France, June 2011.

[GAA 2012] S. GAALLOUL, X. H. B Le, B. Delinchant, F. Wurtz, S. Ploix, «Architecture à composants de co-simulation appliquée au couplage de l'enveloppe thermique du bâtiment au comportement de l'utilisateur» , IBPSA France, Chambéry, Juin 2012.

[GEAR 1984] C. W. Gear, D. R. Wells, «Multirate linear multistep methods». BIT numerical mathematics, volume 24 numéro 4, pp. 484–502, 1984.

[GEO 1998] M. Georgeff et al., «The Belief-Desire-Intention Model of Agency», Agents Theories, Architectures and Languages ATAL, Paris, France, Juillet 1998.

[GILB 2004] N. Gilbert, «Agent-based social simulation: dealing with complexity», disponible sur <http://cress.soc.surrey.ac.uk/resources/ABSS%20-%20dealing%20with%20complexity-1-1.pdf>, Decembre 2004.

[GIEC 2007] Groupe d'experts Intercontinental sur l'Evolution du Climat, Bilan 2007 des Changements climatiques. Disponible sur : http://www.ipcc.ch/pdf/assessment-report/ar4/syr/ar4_syr_fr.pdf

[GOD 1992] K. Gödel, «On Formally Undecidable Propositions Of Principia Mathematica And Related Systems», Dover reprint of the 1962 Basic Books edition, 1992.

[GONZ 2010] F. Gonzalez, M. A. Naya, A. Luaces, M. Gonzalez, «On the effect of multirate co-simulation techniques in the efficiency and accuracy of multibody system dynamics», *Multibody system dynamics journal*, volume 25, numéro 4, pp. 461-483, 2010.

[GRE 2009] Loi Grenelle 1, accessible sur <http://www.legrenelle-environnement.fr/Loi-Grenelle-1,74.html>, Juin 2009

[GRAV 2001] A. Gravouil, A. Combescure, A., «Multi-time-step explicit-implicit method for non-linear structural dynamics», *International Journal for Numerical Methods in Engineering*, volume 50, pp. 199–225, 2001, 2001.

[GU 2011] Lixing Gu, «Advancement of EnergyPlus and its coupling with Champs-whole building», *The 8th International Forum and Workshop on Combined Heat, Air, Moisture and Pollutant Simulations*, Chine, March 2011.

[GUI 2007] D. Guihal, «Modélisation en langage VHDL-AMS des systèmes pluridisciplinaires», thèse de doctorat de l'université de Toulouse III, spécialité micro-électronique, Mai 2007.

H

[HA 2007] L. Duy Ha, «Un système avancé de gestion d'énergie dans le bâtiment pour coordonner production et consommation». Thèse de doctorat INPG, 2007.

[HAM 2012] M. Hammadi, «Contribution à l'intégration de la modélisation et la simulation multi-physique pour la conception des systèmes mécatroniques», thèse de doctorat de l'école centrale de Paris, spécialité génie industrielle, Janvier 2012.

[HARD 2008] C. Hardebolle, «Composition de modèles pour la modélisation multi-paradigme du comportement des systèmes», Thèse de doctorat de l'Université Paris-Sud XI Orsay, spécialité informatique, 2008.

[HEN 2000] J. L. M. Hensen, J. A Clarke, «Integrated simulation for HVAC performance prediction: State of the art illustration», *Proceedings of Int. ASHRAE/CIBSE Conf*, Ireland, 2000.

[HEN 2004] J. L. M. Hensen, E. Djunaedy, M. Radošević, A. Yahiaoui, «Building performance simulation for better design: Some issues and solutions», *Proceedings of 21st Conference on Passive and Low Energy Architecture*, 2004.

[HEN 2012] J. Hensen, «Building performance simulation: current state and challenges», *Expert meeting on Evaluating and Modelling Near-Zero Energy Buildings; are we ready for 2018?*, Glasgow, Janvier 2012.

[HILL 1986] M. Hillestad, T. Hertzberg, «Dynamic simulation of chemical engineering systems by the sequential modular approach», *Computers & Chemical Engineering journal*, volume 10, numéro 4, pp. 377–388, 1986.

[HYS] IEE CSS Technical Committee on Hybrid Sytems
<http://www.dii.unisi.it/hybrid/ieec/index.php?p=scope>.

I

[IEA 2011] Key world energy statistics, IEA (International Energy Agency), technical report, 2011: http://www.iea.org/textbase/nppdf/free/2011/key_world_energy_stats.pdf.

[IEEE 1999] IEEE DASC, «IEEE standard VHDL analog and mixed-signal extensions», December 1999 (revise May 2007).

[INC 1996] F.P. Incropera, D. P. Dewitt, «Fundamentals of Heat and Mass Transfer», 4^{ème} édition, John Willey & sons, New York, 1996.

J

[JAN 1999] M. Janak, «Coupling building energy and lighting simulation», Proceedings of the 5th International IBPSA Conference, Kyoto, Japan, 1999.

[JAN 2009] Y. Jannot, «Transferts thermiques », cours de 2^{ème} année de l'école des Mines de Nancy, 2009, disponible sur <http://www.thermique55.com/principal/thermique.pdf>.

[JARD 2010] A. Jardin, «Contribution à une méthodologie de dimensionnement des systèmes mécatroniques: analyse structurelle et couplage à l'optimisation dynamique», Institut National des Sciences Appliquées de Lyon, 2010.

[JON 2007] D. M. Jones, J. Watton, K. J. Brown, «Comparison of black-, white-, and grey-box models to predict ultimate tensile strength of high-strength hot rolled coils at the Port Talbot hot strip mill», Journal of Materials: Design and Applications, vol. 221, January 2007.

K

[KAS 2011] A. Kashif., X. H. B. Le, J. Dugdale, S. Ploix, «Agent based framework to simulate inhabitants' behaviour in domestic settings for energy management», ICAART, Rome, Italie, 2011.

[KEI 2002] W. Keilholz, «TRNSYS World-wide», IBPSA news, volume 12, numéro 1, 2002.

[KAR 2000] D.C. Karnopp, D.L. Margolis, R.C. Rosenberg, «SystemDynamics, Modeling and simulation of mechatronic systems», John Wiley and Sons Inc., 2000.

[KOS 2006] R. Kossel, W. Tegethoff, M. Bodmann, N. Lemke, «Simulation of complex systems using Modelica and tool coupling», Modelica Conference, Vienne, Autriche, Septembre 2006.

[KAYS 1984] W. M. Kays, A. L. London, «Compact Heat Exchangers », McGraw-Hill, 1984.

[KOF 2009] J. Koffi, «Analyse multicritère des stratégies de ventilation en maisons individuelles», thèse de doctorat de l'université de la Rochelle, spécialité génie civile, Juillet 2009.

[KEIL 2005] W. Keilholz, P. Sette, «Les évolutions de TRNSYS – la version 16», Journée thématique SFT-IBPSA, Mars 2005.

[KUS 1999] T. Kusuda, «Early history and future prospects of buildings system simulation», Building Simulation Conference, Japan, 1999.

[KUZ 2010] D. Kuzmin, «A Guide to Numerical Methods for Transport Equations», disponible sur <http://www.mathematik.uni-dortmund.de/~kuzmin/Transport.pdf>, 2010 :

[KYO 1998] Protocole de Kyoto à la convention-cadre des nations unies sur les changements climatiques, Nations Unies, 1998. Consultable sur: <http://unfccc.int/resource/docs/convkp/kpfrench.pdf>.

L

[LAR 2004] J. Walter Larson et al., «Components, the Common Component Architecture, and the Climate/Weather/Ocean Community», Proceedings of the 20th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, Janvier 2004.

[LE 2008] Le K., «Gestion optimale des consommations d'énergie dans les bâtiments», Thèse de doctorat de l'INPG, spécialité génie électrique, 2008.

[LE 2010] X. H. B. Le et al., «Simulation inhabitant behaviour to manage energy at home», la conférence IBPSA-France , Moret sur loing, France, 2010.

[LEMAR 2000] Philippe Le Marrec, «Conception multi niveaux dans un flot de conception multilanguage», thèse de doctorat de l'INPG, spécialité microélectronique, Juin 2000.

[LI 2009] Z. Li, L. Zheng, H. Zhang, «Modelling and simulation of PDE problems in Modelica», Journal international Materials and Structural Integrity, volume 3, numéro 4, pp 318-331, 2009.

[LID 1996] M. W.Liddament, «A guide to energy efficient ventilation», Air Infiltration and Ventilation Center, Coventry, pp 274, 1996.

[LU 2009] W. Lu, «Le délestage optimal pour la prévention des grandes pannes d'électricité», thèse de doctorat INPG, 2009.

M

[MAC 1972] I.L. Maclaine-cross, P. J. Banks, P. J, «Coupled heat and mass transfer in regenerators-prediction using analogy with heat transfer», Journal international Heat Mass Transfer, Elsevier, volume 15, issue 6, pp 1225–1242, 1972.

[MAR 1966] S. A. Marsshall, «An approximate method for reducing the order of a linear system Control», volume 10, pp. 642–643, 1966.

[MAR 2001] D. Marples, P. Kriens, «The Open Services Gateway Initiative: An Introductory Overview», IEEE Commun.Mag, 2001.

[MAR 2002a] D. Marchio, P. Stabat, «Echangeur rotatif non hygroscopique», Ecole des Mines de PARIS, Centre d'Energétique, 2002.

[MAR 2002b] D. Marchio, O. Morisot, «PREPHC HCSIMOL Modélisation simplifiée d'une batterie chaude dans l'optique d'un calcul des consommations d'énergie d'une installation de climatisation dans un bâtiment tertiaire», Ecole des Mines de PARIS, Centre d'Energétique, 2002.

[MAU 2000] P. M. Maurer, «Components: What if they gave a revolution and nobody came», IEEE Software, pp 28-34, June 2000.

[MAZ 2009] L. Mazzarella, M. Pasini, «Building energy simulation and object-oriented modelling: review and reflections upon achieved results and further developments», Building Simulation conference, Glasgow, 2009. [MERD 2011] A. Merdassi et al. , « Etat de l'art sur les méthodes de modélisation pour les infrastructures critiques interdépendantes », WISG, Troyes France, 2011.

[MAZ 2011] J. Mazet, «La loi Nome: contenu et impact», présentation EDF, disponible sur http://www.cci79.com/petit-dej/doc/2011_05-presentation-EDF.pdf, Mai 2011.

[MCC 1960] J. McCarthy et al., «LISP I Programmers Manual », Artificial Intelligence Group, Mars 1960.

[MEDAD 2006] Ministère de l'écologie, de l'énergie, du développement durable et de la mer, Le plan climat de la France, «Mise en oeuvre du Grenelle Environnement», 2006, disponible sur : http://www.developpement-durable.gouv.fr/IMG/pdf/09003_PLAN_CLIMAT.pdf.

[MEI 1997] T.D. Meijler and O. Nierstrasz, «Beyond Objects: Components, Cooperative Information Systems: Current Trends and Directions», M.P. Papazoglou and G. Schlageter (Eds.), pp. 49-78, Academic Press, November 1997.

[MEM 2004] RTE, «Mémento de la sûreté du système électrique», Edition 2004, www.rte-france.com.

[MEN 2007] C. Ménézo, J.B. Saulnier, D. Lincot, et G. Guarracino, «Energy, domotics, materials welcome to the home of the future (2007)», CNRS international magazine, 2007.

[MER 2005] S. C. Marsell, D. V. Pynadath, S. J. Read, «PsychSim: Agent-based modeling of social interactions and influence», International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, Juillet - Aout 2005.

[MIC 2009] J. Michaelsen, J. Eiden, «HumanComfort Modelica-library thermal comfort in buildings and mobile applications» Modelica conference, Italie, 2009.

[MICH 2004] F. Michel, «Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents», thèse de doctorat de l'Université des Sciences et Techniques du Languedoc, 2004.

[MIL 2010] F. Milhem, «Monitoring d'un système de contrôle et de gestion de consommation électrique», rapport de stage d'alternance, IUT 1 de Grenoble, Septembre 2010.

[MIS 2012] R. Missaoui, «Gestion énergétique optimisée pour un bâtiment intelligent multi-source multi-charges: Différents principes de validations», Thèse de doctorat de l'université de Grenoble, spécialité génie électrique, Juillet 2012.

[MOR 1999] O. Morisot, D. Marchio, «HEATEX Echangeur de Chaleur: modèle NUT- ϵ en enthalpie ou en température de l'HVAC2 Toolkit», Ecole des Mines de PARIS, Centre d'Energétique, 1999.

[MORA 2003] L. Mora, «Prédiction des performances thermo-aérauliques des bâtiments par association de modèles de différents niveaux de finesse au sein d'un environnement orienté objet », Thèse de doctorat de l'UFR Sciences Fondamentales et Sciences pour l'Ingénieur Spécialité Génie Civil, 2003

[MORA 2009] L. Mora, «Etat de l'art en termes de modèles et d'outils de simulation », livrable Projet SIMBIO, 2009.

N

[NIKO 2002] T. Nikolaou, D. Kolkotsa, G. Stavrakakis, «Intelligent Buildings: The Global Framework», pp. 7-25, 2002.

[NIKO 2004] T. Nikolaou, D. Kolkotsa, G. Stavrakakis, «HandBook for Intelligent Building», 2004: <http://www.ibuilding.gr/handbook/index.html>

O

[OLS 2005] H. Olsson, «External Interface to Modelica in Dymola», Proceedings of the 4th International Modelica Conference, Hamburg, March 7-8, 2005.

P

[PAN 2011] X. Pang et al. , «Real time building energy simulation using EnergyPlus and the building virtual test bed», the 12th IBPSA conference, Sidney, Australie, Novembre 2011.

[PAUL 1999] C.A.H. Paul «The treatment of derivatives discontinuities in differential equations», Numerical analysis report, Manchester Centre for Computational Mathematics, Department of Mathematics, 1999.

[PHAM 2011] P.Pharm-Quang, «Modélisation magnéto-mécanique d'un nano commutateur. Optimisation sous contraintes de fiabilité par dérivation automatique des programmes en Java », thèse de doctorat de l'université de Grenoble, Octobre 2011.

[PHAM 2012] -P. Pham-Quang, B. Delinchant. «Java Automatic Differentiation Tool Using Virtual Operator Overloading In Recent Advances in Algorithmic Differentiation», S. Forth et al. (eds.), pp 241-250, Springer; 2012.

[PEUP 1990] B. Peuportier, I. Blanc-Sommereux, «Simulation tool with its expert interface for the thermal design of multizone buildings» International Journal of Sustainable Energy, vol 8, issue 2, p.109–120, 1990.

[PLOI 2012] S. Ploix, «Energy management in dwellings:simulating complex behaviours of occupants ?», école de printemps SIMUREX 2012, Corse, Avril 2012.

[POQ 2008] Poquet G, Dujin A., «Pour les ménages, la recherche du confort prime encore sur les économies d'énergie», Crédoc N°210, Mars 2008, disponible sur : <http://www.credoc.fr/pdf/4p/210.pdf> .

[POP 2008] A. Pop, «Integrated Model-Driven Development Environments for Equation-Based Object-Oriented Languages», Thèse de doctorat de l'université de Linköping, suède, Juin 2008.

[PREB 2007] rapport final de projet PREBAT réalisé par l'ADEME/PUCA/CSTB, «Comparaison internationale Bâtiment et énergie : Rapport final», disponible sur <http://www.prebat.net/IMG/pdf/a-total-2008.pdf> 2007.

[PRED 2012] la plaquette de Predis, disponible sur la lien www.g2elab.grenoble-inp.fr/servlet/com.univ.collaboratif.util.LectureFichier?CODE_FICHER=1236783925750&ID_FICHE=131526.

R

[RAZ 2006] [3] H. Razik, «La machine asynchrone à vitesse variable 1», Lavoisier, 2006.

[REG 2003] J. Regnier, «Conception de systèmes hétérogènes en Génie Electrique par optimisation évolutionnaire multicritère», Thèse de doctorat d'INP de Toulouse, Décembre 2003.

[REID 2004] P. Riederer, W. Keilholz, M. Kummert, «Type 155 – un nouveau type TRNSYS pour coupler TRNSYS et Matlab», TRNSYS Userdays, CSTB, Sophia Antipolis, 9/10 mars 2004.

[REID 2009] P. Riederer, W. Keilholz, V. Ducreux, «Coupling of TRNSYS with SIMULINK- a method to automatically export and use TRNSYS models within SIMULINK and vice versa», Building Simulation, Glasgow, Royaume-Uni, 2009.

[REN 2011] R. Renier, R. Chenouard, «De SysML à Modelica : aide à la formalisation de modèles de simulation en conception préliminaire» 12ème Colloque National AIP PRIMECA, Le Mont Dore, Mars- avril 2011.

[REZ 2010] A. Rezgui, «Modélisation VHDL-AMS unifiée en support du cycle en V de conception Application aux filtres ADSL», rapport de master de recherche, université de 7 Novembre de Carthage, Tunis, Decembre 2010.

[REZ 2011] A. Rezgui et al, «Unified Modelling technique using VHDL-AMS and software components», ELECTRIMACS, Cergy Pontoise, France, June 2011.

[REZ 2012] A. Rezgui, «Interopérabilité de modèles dans le cycle de conception des systèmes électromagnétiques via des supports complémentaires: langage VHDL-AMS et composants logiciels ICAr», soutenance prevue Octobre 2012.

[REZ 2012b] A. Rezgui, «VHDL-AMS to Support DAE-PDE Coupling and Multilevel Modeling», IEEE transactions on magnetics, vol 48, No 2, 2012.

[ROB] X. Roboam, S.Astier, «Graphes de liens causaux pour systèmes à énergie renouvelable», Techniques de l'Ingénieur, D 3970.

[ROUX 1984] J.J. Roux, «Proposition de modèles simplifiés pour l'étude du comportement thermique des bâtiments», Thèse de doctorat, INSA Lyon, 1984.

[RT 2005] Direction générale de l'Urbanisme, de l'Habitat et de la Construction, «Réglementation Thermique 2005 : des bâtiments confortables et performants», version Octobre 2006, http://www.developpement-durable.gouv.fr/IMG/pdf/rt2005_version09102006.pdf.

[RT 2012] Ministère de l'Écologie, du Développement durable, des Transports et du Logement, «Réglementation thermique 2012 : un saut énergétique pour les bâtiments neufs», Avril 2011, consultable sur : http://www.developpement-durable.gouv.fr/IMG/pdf/DGALN_plaquetteRT2012_avril2011.pdf.

[RTE] : Réseau de Transport d'Electricité, «Maîtriser la consommation d'électricité», consultable sur le lien : <http://www.rte-france.com/fr/developpement-durable/maitriser-sa-consommation-electrique/maitriser-la-consommation-d-electricite>.

[RTE 2011] Réseau de Transport d'Electricité, «Bilan Prévisionnel de l'équilibre offre - demande d'électricité en France», édition 2011.

[RUEL 2002] Régis Ruelland, «Apport de la co-simulation dans la conception de l'architecture des dispositifs de commande numérique pour les systèmes électriques», thèse de doctorat de l'INPG, spécialité génie électrique, Septembre 2002.

[RUS 2005] M. Russel, M. Sherman, A. Rudd, «Review of Residential Ventilation Technologies», Lawrence Berkeley National Laboratory LBNL, august 2005.

S

[SAG 2011] C. Sagerschnig et al. , «Co-simulation for building controller development: the case study of a modern office building», CISBAT, Lausanne, Suisse, Septembre 2011.

[SAL 2000] L. Saldamli, P. Fritzson, «Object oriented modeling with Partial Differential Equation», workshop Modelica, Sweden, Octobre 2000.

- [SAL 2002] L. Saldamli, P. Fritzon, B. Bachmann, «Extending Modelica for Partial Differential Equations», 2^{ème} conférence internationale de Modelica, Allemagne, Mars 2002.
- [SAL 2005] T. Salomon, R. Mikolasek, B. Peuportier, «Outil de simulation thermique du bâtiment, Comfie» Journée thématique SFT-IBPSA, 2005
- [SCH 2003] C. Schneeweiss, «Distributed Decision Making», Springer 2003.
- [SEG 1982] G. Segulier et F. Notelet, «Electronique industrielle», Tec et doc Lavoisier, 1982.
- [SEMP 2011] A. Sempey, L. Mora, «Détermination du point de fonctionnement du réseau aéronautique», rapport du projet SIMINTHEC, Décembre 2011.
- [SHA 1970] D. F., Shanno, «Conditioning of quasi-Newton methods for function minimization», Mathematics of Computation, 24:145-160, 1970.
- [SIM 2001] Guide d'utilisation de Simulink, «Simulink Model-Based and System-Based Design: writing S-function», version 4, Juin 2001.
- [SIM 2010] «Tutorial école SIMUREX: Mise en œuvre de l'approche composant logiciel avec le framework cades V2.2», Simurex 2010, Corse, France.
- [SIMINTHEC 2009] projet ANR-Habisol, «SIMulation et outils logiciels INTERopérables pour la gestion Thermique et EleCTrique dans les bâtiments», 2009-2012, présentation disponible sur <http://www.prebat.net/IMG/pdf/ewurtz.pdf>.
- [SJO 2009a] M. Sjölund, «Bidirectional External Function Interface Between Modelica/MetaModelica and Java», Thèse de master de Linköpings, Aout 2009.
- [SJO 2009b] M. Sjölund, P. Fritzon, «An OpenModelica Java External Function Interface Supporting MetaProgramming», Proceedings of the 7th Modelica Conference, Como, Italy, Septembre 2009.
- [SOC 2009] T. Solcia, «Multidisciplinary Multirate Co-simulations in Multibody Dynamics», thèse de l'école polytechnique de Milano, spécialité aéronautique, 200-2009.
- [SOES 2010] Service de l'Observation et des Statistiques, Bilan énergétique de la France pour 2009, Cité dans le Rapport Bâtiment de l'ADEME 2010, disponible sur <http://www2.ademe.fr/servlet/getBin?name=D199DB4F63EC2F0B02A706671367CC231302179711154.pdf>.
- [SOLE] J. Sole, «Thermal Inertia in Buildings», workshop sur l'inertie thermique, consultable sur http://www.telefonica.net/web2/josepsolebonet/index_archivos/Workshop_Thermal_Inertia.pdf.
- [SPEC 2010] Les spécifications du langage Modelica, «Modelica - A Unified Object-Oriented Language for Physical Systems Modeling», Mars 2010.
- [SZY 1998] C. Szypersky, «Component Software - Beyond Object-Oriented Programming», Addison-Wesley, 1998.

T

- [THEI 2009] I. Their, «Contribution au développement de modèles pour l'électronique de puissance en VHDL-AMS», thèse de doctorat de l'université de Lyon, spécialité génie électrique, Janvier 2009.

[THIE 2008] S. THIERS, «Bilans énergétiques et environnementaux de bâtiments à énergie positive», thèse de doctorat de l'Ecole des Mines de Paris, spécialité énergétique, Novembre 2008.

[TIL 2001] M.M. Tiller, «Introduction to Physical Modeling with Modelica, Norwel», Mass.:Kluwer Academic Publisher, 2001.

[TRCK 2006] M. Trcka, J.L.M Hensen, «Model and tool requirements for co-simulation of building performance», proceedings of the 15th IASTED (Interfational conference on Applied Simulation and Modelling), pp.7, June 2006.

[TRC 2008] Marija Trcka, «Co-simulation for Performance Prediction of Innovative Integrated Mechanical Energy Systems in Buildings», Thèse de doctorat de l'Université technique d'Eindhoven, 2008.

[TRN 2007] Guide de programmation de TRNSYS «TRNSYS 16: Programmer's Guide», volume 8, Février 2007.

[TROC 2009] M. TROCMÉ, «Aide aux choix de conception de bâtiments économes en énergie», thèse de doctorat de l'Ecole des Mines de Paris, spécialité énergétique, Novembre 2009.

U

[UFE 2010] Union Française de l'électricité, «Electricité 2030 : quels choix pour la France », disponible sur http://www.ufe-electricite.fr/IMG/pdf/brochure_synthese_ufe_fr_bd.pdf, 2010.

V

[VER 2012] - F. Verdière, A. Rezgui, S. Gaaloul, B. Delinchant, L. Gerbaud, F. Wurtz and X. Brunotte, «Modelica models translation into Java components for optimization and DAE solving using automatic differentiation», IEEE UKSim2012, 14th International Conference on Computer Modelling and Simulation, 28 - 30 March 2012, Cambridge, UK. pp.340-344.

[VUOL 1999] M. Vuolle, A. Bring, P. Sahlin, «An NMF based model library for building thermal simulation », Building Simulation Conference, Kyoto, Japan, 1999.

W

[WAR 2011] G. Warkozek, « Génération automatique de problèmes d'optimisations pour la conception et la gestion des réseaux électriques des bâtiments intelligents multi-sources multi-charges », Thèse de doctorat de l'Université de Grenoble, spécialité génie électrique, Septembre 2011.

[WEB_AUT] <http://www.autosar.org/>.

[WEB_BCV] simulationresearch.lbl.gov/bcvtb.

[WEB_BRA] <http://www.agentisolutions.com/products/index.htm>.

[WEB_COM] <http://www.izuba.fr/logiciel/pleiadescomfie>.

[WEB_DIM] <http://www.dimocode.com/>.

[WEB_DYM] <http://www.3ds.com/products/catia/portfolio/dymola/overview/>.

[WEB_EPE] <http://www.epexspot.com/fr/>.

- [WEB_FMI] <http://www.modelisar.com/index.html>.
- [WEB_HQE] <http://assohqe.org/hqe/spip.php?rubrique9>.
- [WEB_IPS] <http://www.ipsos.fr/>.
- [WEB_JNI] <http://java.sun.com/docs/books/jni/html/intro.html#1811>.
- [WEB_LIB] <https://modelica.org/libraries/Modelica>.
- [WEB_LIN] <http://linky.erdfdistribution.fr/>.
- [WEB_MAT] <http://www.mathworks.fr/products/matlab/>.
- [WEB_MOD] <https://modelica.org/>.
- [WEB_PMAT] <http://www.dimocode.com/alfresco/cecm/pack/viewSpace?spaceId=ae1c0b71-f0d2-467a-8236-61e85f93ec33&content=/descriptor/default>.
- [WEB_PTO] <http://ptolemy.eecs.berkeley.edu/ptolemyII/>.
- [WEB_RAT] <http://www.mathworks.fr/help/toolbox/simulink/slref/ratetransition.html>.
- [WEB_RTE] RTE: Réseau de Transport de l'Electricité, <http://www.rte-france.com/fr/>.
- [WEB_SFU] <http://www.mathworks.fr/help/toolbox/simulink/sfg/f6-151.html>.
- [WEB_SIM] <http://boutique.cstb.fr/fr/simbad.html>.
- [WEB_TRN] <http://sel.me.wisc.edu/trnsys/features/features.html>.
- [WEB_VHD] <http://www.vhdl.org/analog>.
- [WEB_ZER] http://en.wikipedia.org/wiki/Zero_crossing.
- [WET 1999a] M. Wetter, « Simulation Model Air-to-Air Plate Heat Exchanger », Laboratoire Nationale de Lawrence Berkley, Janvier 1999.
- [WET 1999b] M. Wetter, « Simulation Model Finned Water-to-Air Coil Without Condensation », LNBL, Janvier 1999.
- [WET 2008] M. Wetter, P. Haves, « A modular building controls virtual test bed for the integration of heterogeneous systems », Proceedings of SimBuild, 3rd National Conference of IBPSA-USA, Bekeley, CA, USA, 2008.
- [WET 2009] M. Wetter, « A Modelica-based model library for building energy and control systems ». Conference Building Simulation, 2009.
- [WET 2011] M. Wetter, « Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed », *Journal of Building Performance Simulation*, volume4, issue 3, pp 185-203, 2011.
- [WET 2012] M. Wetter, J. Hensen, C. van Treeck, « New generation computational tools for building and community energy systems based on the Modelica and Functional Mockup Unit standards », IEA proposal.

[WINK 1993] F. Winkelmann et al., « DOE-2 Supplement », Version 2.1E, Springfield, Lawrence Berkeley, 1993.

[WU 2007] C. Wu, C. Liao, L. Fu, « Service-Oriented Smart Home Architecture based on OSGi and Mobile Agent Technology », IEEE Transactions On Systems, Man, And Cybernetics, 2007.

[WUR 2008] F. Wurtz et al, « Vers de nouvelles approches pour la capitalisation des modèles pour la simulation et l'optimisation : l'expérience du projet DIMOCODE », Numelec, Belgique, 2008.

[WUR 1992] F. Wurtz, « Une nouvelle approche pour la conception sous contraintes de machines électriques », Thèse de doctorat de l'INPG, spécialité génie électrique, Mars 1992.

Y

[YAMA 2005] S. Yamamoto, K. Sumi, E. Nishikawa, T. Hashimoto, « an Operating Method using Prediction of Photovoltaic Power for a Photovoltaic – Diesel Hybrid Power Generation System », Electrical Engineering in Japan, Vol.151, N°3, 2005.

Z

[ZAY 2011] C. Zayane, « Identification d'un modèle de comportement thermique de bâtiment à partir de sa courbe de charge », Thèse de doctorat de l'école des mines de Paris, Janvier 2011.

[ZEIG 2000] B.P. Zeigler, H. Praehofer, T.G. Kim, « Theory of Modeling and Simulation », Academic Press, Inc, 2000.

[ZIM 2009] D. Zimmer, « Module-Preserving Compilation of Modelica Models », Proceedings 7th Modelica Conference, Como, Italie, Septembre 2009.

ANNEXES

Annexe 1: Structure de G-HomeTech

G-HomeTech [HA 2007] [ABR 2008] est un système de gestion de l'énergie. Il s'agit d'un système complexe rassemblant plusieurs fonctions, issu des travaux de Stéphane PLOIX, chercheur au laboratoire GSCOP²⁹. Il est composé de trois couches [MISS 2012] (Figure 115):

- anticipative permettant la prédiction des acteurs internes (usages, équipements) et externes (coûts, météo),
- réactive pour la correction et l'ajustement afin de respecter les anticipations,
- et une couche locale au niveau des équipements pour la prise des consignes.

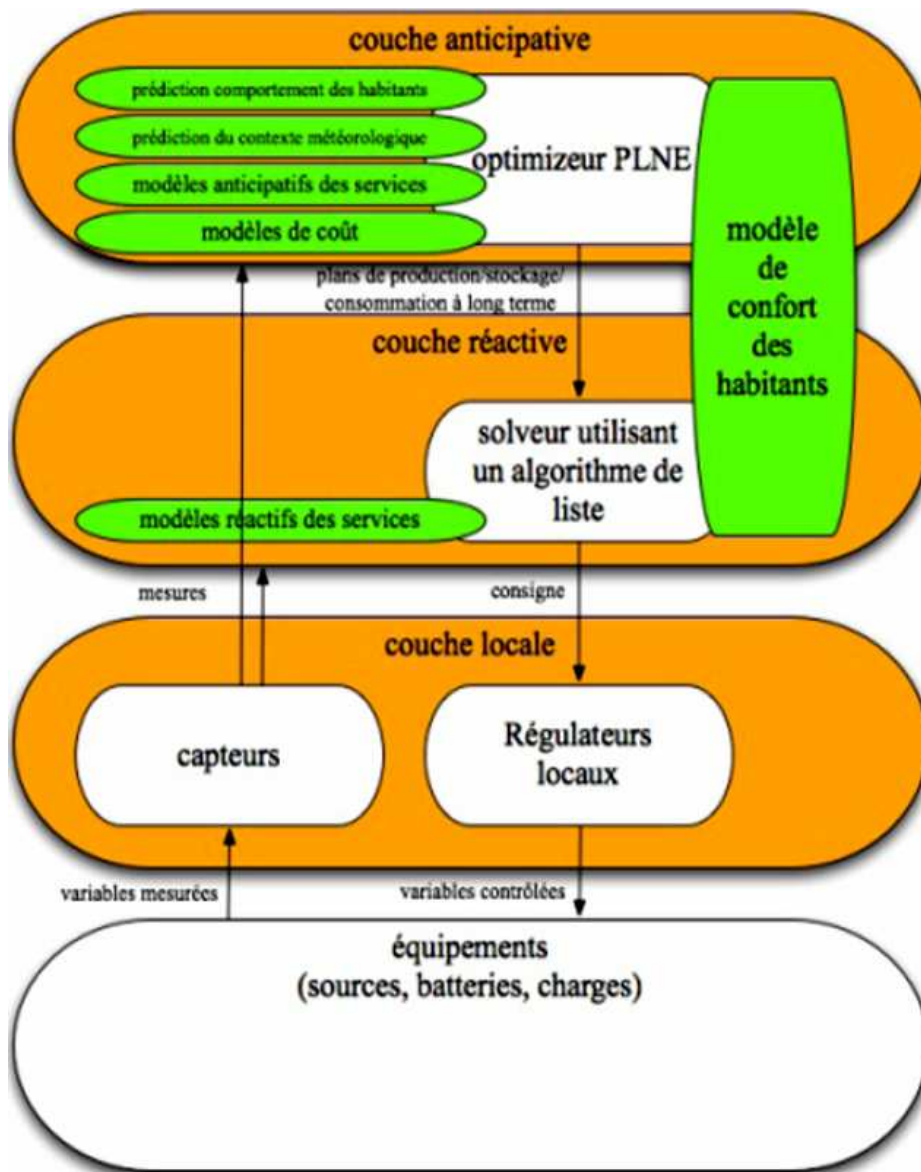


Figure 115. Le système multi-couches de G-HomeTech

²⁹ GSCOP: Laboratoire des Sciences pour la Conception, l'Optimisation et la Production de Grenoble : <http://www.g-scop.grenoble-inp.fr/>

Annexe 2: Présentation générale de TRNSYS

TRNSYS est un programme de simulation dynamique de bâtiments et des systèmes. Il est composé d'une interface graphique «Simulation Studio », d'un cœur de calcul et d'une bibliothèque très riche de composants appelés « Type ». L'interface graphique permet de générer les fichiers d'entrée pour le solveur TRNSYS à partir d'une description graphique en fournissant les caractéristiques du bâtiment et des composants des systèmes ainsi que la manière dont ils sont connectés.

Il existe environ 50 familles de composants génériques standard [KEIL 2005] permettant de simuler les bâtiments (mono - ou multi zonal), les systèmes de chauffage et de climatisation, les régulateurs, des composants électriques, des échangeurs de chaleur, des systèmes à hydrogène, des systèmes solaires thermiques et photovoltaïques, des systèmes de stockage d'énergie thermique et électrique...

Grâce à sa modularité, TRNSYS est extrêmement flexible pour la modélisation d'une grande variété des systèmes énergétiques de différents niveaux de complexité. En effet, il permet, non seulement, de constituer un système énergétique de plusieurs composants de sa bibliothèque (drag and drop) mais aussi il permet l'ajout et l'utilisation de nouveaux composants.

Bien que TRNSYS offre une large possibilité de simulation des systèmes énergétiques, il reste moins précis que COMFIE en ce qui concerne la modélisation de l'enveloppe.

Annexe 3: Le plan du 1^{er} étage de Predis

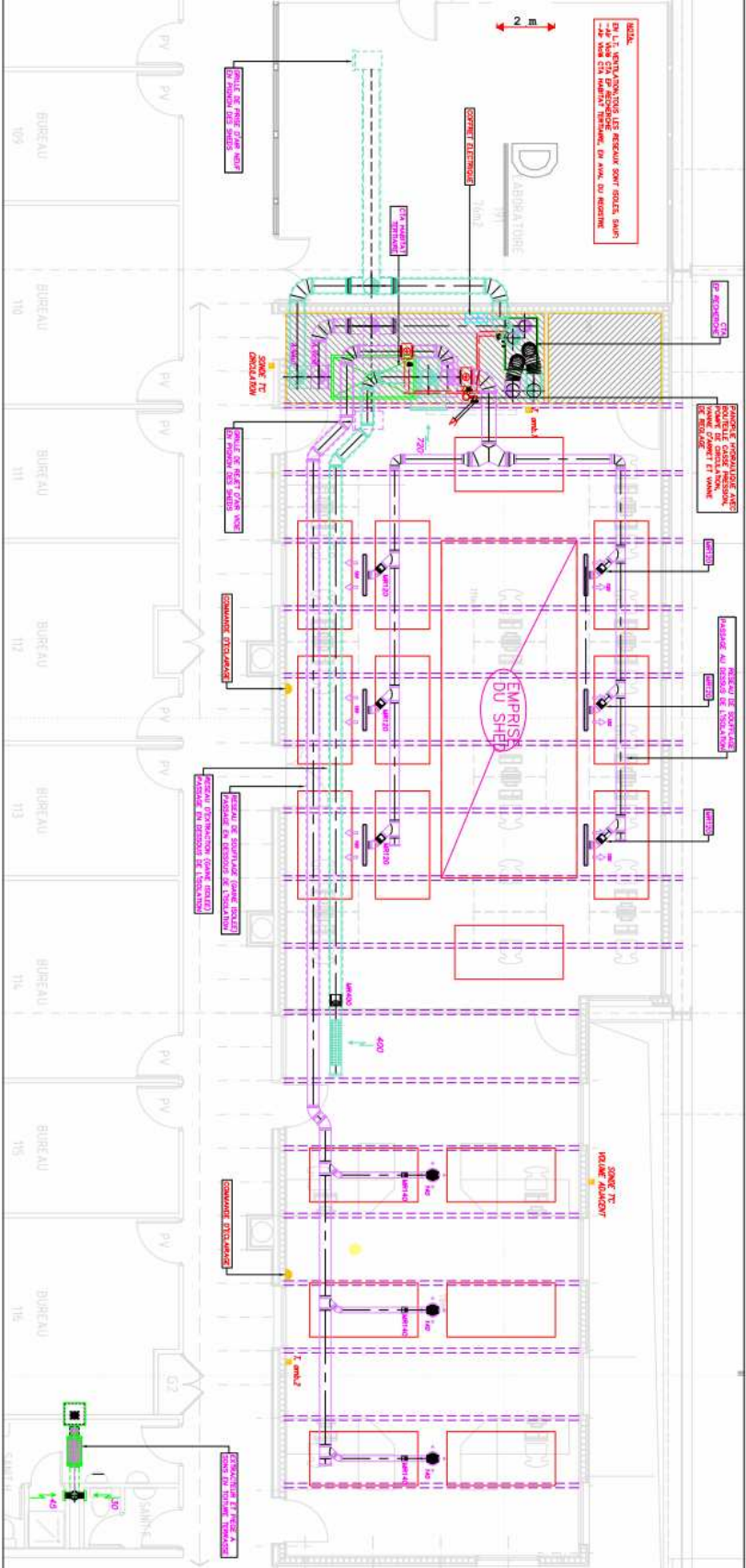


Figure 116. Le plan de chauffage et ventilation du 1^{er} étage de Predis

Annexe 4: Principales étapes de la construction du modèle de l'enveloppe thermique dans COMFIE

Etude par zone du bâtiment :

Dans un premier temps, chaque zone du bâtiment est décomposée en mailles sur lesquelles un bilan thermique est établi en supposant la température uniforme. Selon la position de la maille (maille d'une paroi côté intérieur, maille intermédiaire, maille la plus externe d'une paroi externe...), des équations d'échange sont établies et qui peuvent être formalisées par le système matriciel suivant pour la zone entière.

$$\left. \begin{aligned} C \cdot \dot{T} &= A \cdot T + E \cdot U \\ Y &= J \cdot T + G \cdot U \end{aligned} \right\}$$

où T est le champ discrétisé des températures des mailles

U le vecteur des sollicitations

Y le vecteur des sorties

C la matrice diagonale des capacités thermiques

A la matrice contenant les termes d'échange entre mailles

E contient les termes d'échange entre mailles et sollicitations

J relie les sorties aux températures des mailles

G relie les sorties aux sollicitations

Le vecteur U des sollicitations contient les puissances internes, la température extérieure, les divers flux solaires (si la zone est en contact avec l'extérieur) et des températures équivalentes qui représentent des sorties des zones adjacentes si elles existent.

Le champ de température obtenu en régime permanent est défini par:

$$T = 0, \text{ soit: } T = -A^{-1} \cdot E \cdot U$$

On peut ainsi décrire le champ de température T comme la somme d'un terme en régime permanent et d'un terme dynamique T_0 : $T = T_0 - A^{-1} \cdot E \cdot U$

En remplaçant T par cette expression dans le système précédant, on obtient:

$$\left. \begin{aligned} \dot{T}_0 &= C^{-1} \cdot A \cdot T_0 + A^{-1} \cdot E \cdot \dot{U} \\ Y &= J \cdot T_0 + (G - J \cdot A^{-1} \cdot E) \cdot U \end{aligned} \right\}$$

Ce système sera plus simple à résoudre si la matrice $(C^{-1} \cdot A)$ est diagonale. La diagonalisation de cette matrice est possible grâce à un changement de base, la nouvelle base étant formée des vecteurs propres de $C^{-1} \cdot A$. Le champ T des températures est transformé en vecteur d'état X par la relation: $T = P \cdot X$ où P est la matrice de passage.

On obtient ainsi un système de la forme :

$$\begin{cases} \dot{X} = F \cdot X + B \cdot \dot{U} \\ Y = H \cdot X + S \cdot U \end{cases}$$

où F est une matrice diagonale dont le ième terme est $-1/z_i$ et z_i étant la ième constante de temps de la zone. Une réduction de l'ordre du modèle est par la suite possible en éliminant les constantes de temps les plus faibles.

Couplage des zones:

Afin de rendre indépendantes les sollicitations pour chaque zone du bâtiment, le vecteur des sollicitations (U) est séparé en deux composantes : un vecteur U des sollicitations extérieures (température extérieure, flux solaires, puissances internes) et un vecteur Y des variables de couplage (contenant les températures équivalentes). Les matrices d'états B et S sont aussi séparées en deux composantes α et β .

La composition du nouveau vecteur U est formalisée ainsi en respectant l'ordre indiqué:

1	Température extérieure	} Puissances des usagers, puissance des appareils électrique...: lieu du couplage avec l'électrique
2	Puissance interne zone 1	
...	...	
$n_{zones}+1$	Puissance interne zone n_{zones}	
$n_{zones}+2$	Flux incident 1	} Flux solaires incidents: calculées par Comfie
...	...	
$nb_{mesures}$	Flux incident $nsoll-1$	

Ensuite les systèmes correspondants à chaque zone sont couplés afin de former le système global (Chapitre2, Sys2) :

$$\begin{cases} \dot{X} = F \cdot X + B^\alpha \cdot \dot{U} + B^\beta \cdot \dot{Y} \\ Y = H \cdot X + S^\alpha \cdot U + S^\beta \cdot Y \end{cases}$$

Annexe 5: Le modèle de la partie électrique de la VMC (moteur et variateur de vitesse) dans Modelica

Le modèle global:

Le modèle global de la partie électrique de la VMC, représenté dans cet annexe est un modèle acausal composant : un variateur de vitesse, une machine asynchrone, un bloc de consigne de vitesse et une charge de couple résistant (Figure 62).

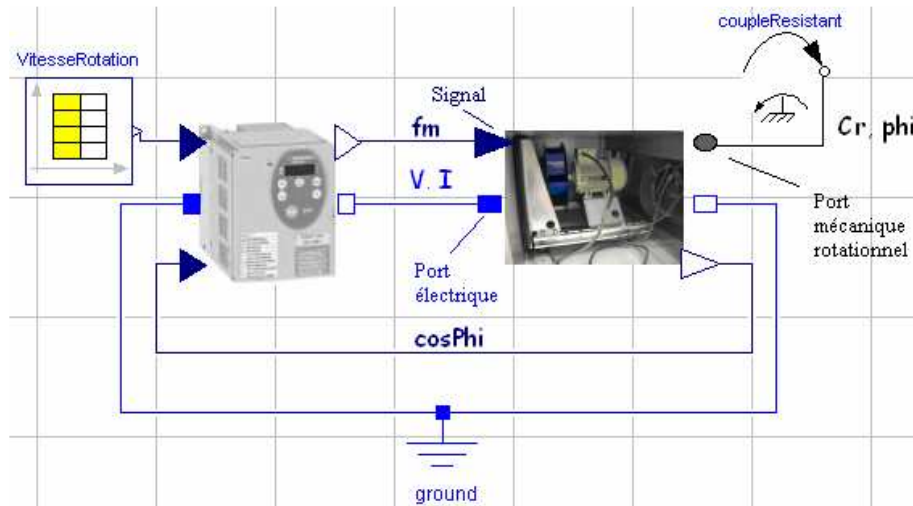


Figure 117. Modélisation acausale de la partie électrique de la VMC dans Modelica

Les codes Modelica correspondants au modèle global ainsi qu'à ces différents éléments sont fournis ici :

```

model VMC_doubleFlux_flanges
  Modelica.Electrical.Analog.Basic.Ground ground;
  MAS_Flange MAS;
  Variateur_pin_block Variateur;
  Modelica.Blocks.Sources.Ramp Vitesse_Rotation(duration=3000,offset=1,startTime=1,height=3000);
  Real Rendementglobal;
  Couple_resistant couple_resistant;
equation
  Rendementglobal=Variateur.Rendement*MAS.RendementMAS;
  connect(MAS.flange, couple_resistant.flange);
  connect(ground.p, Variateur.p);
  connect(Variateur.n, MAS.p);
  connect(MAS.n, ground.p);
  connect(Vitesse_Rotation.y, Variateur.u[1]);
  connect(MAS.u, Variateur.y);
  connect(Variateur.u[2], MAS.y);
end VMC_doubleFlux_flanges;

```

Le modèle de la machine asynchrone

```

model MAS_Flange
  extends Modelica.Electrical.Analog.Interfaces.OnePort;
  extends Modelica.Mechanics.Rotational.Interfaces.PartialOneFlangeAndSupport;
  extends Modelica.Blocks.Interfaces.SISO;
  parameter Real fr=0.0012;
  parameter Real Lm=0.795;
  parameter Real Nr=0.041;
  parameter Real pnb=1;
  parameter Real Rm=17090;

```

```

parameter Real Rr=7.106;
parameter Real Rs=13.909;
Real f=u; //fréquence
Real cosP; // Facteur de puissance
Real Vs;
Modelica.SIunits.AngularFrequency Or;
Modelica.SIunits.Torque Cu; // Couple utile
Real RendementMAS; // Rendement du moteur
Real Is; // Courant statorique
//Variables intermédiaires
Real g; // Glissement
Real ws; // Pulsation statorique
Real Xr; // Réactance statorique
Real Xm; // Réactance magnétisante
Real R; // Résistance magnétisante équivalent
Real X; // Réactance magnétisante équivalent
Real Rtotale; // Résistance équivalent totale
Real Xtotale; // Réactance équivalent totale
Real Ir; // Courant rotorique
Real Io; // Courant magnétisant
Real Pem; // Puissance électromécanique
Real Cem; // Couple électromécanique
Real Pmec; // Puissance mécanique
Real Cfr; // Couple de résistance mécanique visqueux
Real DPmec; // Pertes mécanique
Real Pu; // Puissance utile
Real Pabs; // Puissance active absorbée
Real DPs; // Pertes joules au stator
Real DPr; // Pertes joules au rotor
Real DPfe; // Pertes fer
Real DP; // Pertes totales
equation
Cu=-flange.tau;
Or=der(-flange.phi);
Vs=-v;
Is=-i;
//CoupleRes=1.25e-7*(Or^2);
//CoupleRes=Cu;
g=(60*f/pnb-Or)/(60*f/pnb); // Glissement
ws=2*3.141529*f; // Pulsation statorique
Xr=Nr*ws; // Réactance statorique
Xm=Lm*ws; // Réactance magnétisante
// Résistance magnétisante équivalent
R=Rm*Xm*Xm/(Rm*Rm+Xm*Xm);
// Réactance magnétisante équivalent
X=Rm*Rm*Xm/(Rm*Rm+Xm*Xm);
// Résistance équivalent totale
Rtotale=Rs+(Rr*R/g-
Xr*X)*(Rr/g+R)/((Rr/g+R)^2+(Xr+X)^2)+(Xr*R+Rr*X/g)*(Xr+X)/((Rr/g+R)^2+(Xr+X)^2);
// Réactance équivalent totale
Xtotale=(Xr*R+Rr*X/g)*(Rr/g+R)/((Rr/g+R)^2+(Xr+X)^2)-(Rr*R/g-Xr*X)*(Xr+X)/((Rr/g+R)^2+(Xr+X)^2);
// Courant statorique
Is=Vs/((Rtotale^2+Xtotale^2)^(1/2));
// Courant rotorique
Ir=((R*R+X*X)^(1/2))*Is/(((R+Rr/g)*(R+Rr/g)+(X+Xr)*(X+Xr))^(1/2));
// Courant magnétisant
Io=((Rr/g*Rr/g+Xr*Xr)^(1/2))*Is/(((R+Rr/g)*(R+Rr/g)+(X+Xr)*(X+Xr))^(1/2));
// Puissance électromécanique
Pem=3*(Rr/g)*(Ir^2);
// Couple électromécanique

```

```

Cem=Pem/(ws/pnb);
// Puissance mécanique
Pmec=3*Rr/g*(1-g)*(Ir^2);
// Facteur de puissance
cosP=Rtotale/(((Rtotale^2)+(Xtotale^2))^(1/2));
// Couple de résistance mécanique visqueux
Cfr=fr*ws*(1-g)/pnb;
// Pertes mécanique
DPmec=Cfr*ws*(1-g)/pnb;
// Couple utile
Cu=Cem-Cfr;
// Puissance utile
Pu=Pmec-DPmec;
// Puissance active absorbée
Pabs=3*Vs*Is*cosP;
// Rendement du moteur
RendementMAS=Pu/Pabs;
// Pertes joules au stator
DPs=3*Rs*(Is^2);
// Pertes joules au rotor
DPr=3*Rr*(Ir^2);
// Pertes fer
DPfe=3*R*(Io^2);
// Pertes totales
DP=DPs+DPr+DPfe;
y=cosP;
end MAS_Flange;

```

Le modèle du variateur

```

model Variateur_pin_block
extends Modelica.Electrical.Analog.Interfaces.OnePort;
extends Modelica.Blocks.Interfaces.MISO(nin=2);
parameter Real Vb=400/(3^(1/2));
parameter Real fb=50;
parameter Real fh=12000;
parameter Real E=Vb*(6^(1/2));
parameter Real Ion_ref=3;
parameter Real Won_ref=1.1e-3;
parameter Real Etest=480;
parameter Real Ioff_ref=3;
parameter Real Woff_ref=1.8e-3;
parameter Real Rigbt=0.35;
parameter Real Vce0=0.9;
parameter Real Id_ref=3;
parameter Real Rdi=0.25;
parameter Real Qrr_ref=4500e-9;
parameter Real Vd0=0.9;
Real Oconsigne=u[1];
Real cosPm=u[2];
Real fm;
constant Real PI=Modelica.Constants.pi;
Real Vm=v;
Real Im=i;
Real Pertes;
Real Rendement;
//protected
Real IK1_IGBT1_moy(start=0);
Real IK1_IGBT1_eff2(start=0);
Real IK1_D1_moy(start=0);

```

```

Real IK1_D1_eff2(start=0);
Real Pertes_commutation_D1(start=0);
Real Pertes_commutation_IGBT1;
Real Perte_conduction_IGBT1;
Real Perte_conduction_D1;
Integer k=1;
Real moy2;
Real moy1;
Real moy3;
Real Won;
Real Woff;
protected
package FonctionsVariateur
function a1
  input Real E;
  input Real t;
  input Real Vm;
  input Real fm;
  output Real a1output;
  constant Real PI=Modelica.Constants.pi;
algorithm
  a1output:=Vm*(2^(1/2))/E*sin(2.0*PI*fm*t)+0.5;
end a1;
function IK1
  input Real E;
  input Real fh=12000;
  input Real t;
  input Real Vm;
  input Real Im;
  input Real fm;
  input Real cosPm;
  output Real IK1out;
  constant Real PI=Modelica.Constants.pi;
protected
  Real Ia;
  Real k;
  Real tmoy;
algorithm
  Ia:=Im*(2^(1/2))*sin(2*PI*fm*t-Modelica.Math.acos(cosPm));
  k:=floor(t*fh);
  tmoy:=(2*k + 1)/(2*fh);
  if (t<(a1(E,tmoy,Vm,fm)/fh+k/fh)) then
    IK1out:=Ia;
  else
    IK1out:=0;
  end if;
end IK1;
function IK1_D1
  input Real E;
  input Real fh=12000;
  input Real t;
  input Real Vm;
  input Real Im;
  input Real fm;
  input Real cosPm;
  output Real IK1_D1;
  constant Real PI=Modelica.Constants.pi;
protected
  Real IK1out;
algorithm

```



```

IK1out:=IK1(E,fh,t,Vm,Im,fm,cosPm);
if (IK1out<0) then
  IK1_D1:=-IK1out;
else
  IK1_D1:=0;
end if;
end IK1_D1;
function IK1_IGBT1
input Real E;
input Real fh=12000;
input Real t;
input Real Vm;
input Real Im;
input Real fm;
input Real cosPm;
output Real IK1_IGBT1out;
constant Real PI=Modelica.Constants.pi;
protected
Real IK1out;
algorithm
IK1out := IK1(
  E,
  fh,
  t,
  Vm,
  Im,
  fm,
  cosPm);
if (IK1out > 0) then
  IK1_IGBT1out := IK1out;
else
  IK1_IGBT1out := 0;
end if;
end IK1_IGBT1;
end FonctionsVariateur;
algorithm
for k in 0:floor(fh/fm) loop
  // Calcul le courant moyen et courant efficace au carré
  // IGBT
  moy2:=Vm*(2.0^(1/2))/E*sin(2.0*PI*fm*((k+0.5)/fh))+0.5;
  moy1:=FonctionsVariateur.IK1_IGBT1(E,fh,(2*k+moy2)/(2*fh),Vm,Im,fm,cosPm);
  // Calcule la valeur moyen du courant passant
  IK1_IGBT1_moy:=IK1_IGBT1_moy+fm*moy1*moy2/fh;
  // Calcule la valeur efficace du courant passant
  IK1_IGBT1_eff2:=IK1_IGBT1_eff2+fm*(moy1^2)*moy2/fh;
  // Diode anti-parallèle :
  // Valeur moyenne de courant passant :
  moy3:=FonctionsVariateur.IK1_D1(E,fh,(2*k+FonctionsVariateur.a1(E,(k+0.5)/fh,Vm,fm))/(2*fh),Vm,Im,fm
,cosPm);
  IK1_D1_moy:=IK1_D1_moy+fm*moy3*moy2/fh;
  // Valeur efficace de courant passant :
  IK1_D1_eff2:=IK1_D1_eff2+fm*(moy3^2)*moy2/fh;
  // Perte en commutation d'un interrupteur
  // Diode anti-parallèle :
  Pertes_commutation_D1:=Pertes_commutation_D1+fm*moy3*Qrr_ref/Id_ref*E;
  //IGBT
  Won:=Won_ref/(Etest*Ion_ref)*E*moy1;
  Woff:=Woff_ref/(Etest*Ioff_ref)*E*moy1;
  Pertes_commutation_IGBT1:=Pertes_commutation_IGBT1+fm*(Won+Woff);
end for;

```

equation

```
fm=0.024*(Oconsigne^0.975);
Vm=0.07*(Oconsigne^0.957);
Perte_conduction_IGBT1=Vce0*IK1_IGBT1_moy+Rigt*IK1_IGBT1_eff2;
// Diode anti-parallèle :
Perte_conduction_D1=Vd0*IK1_D1_moy+Rdi*IK1_D1_eff2;
//Pertes et rendement du variateur
Pertes = 6.0 * (Perte_conduction_IGBT1 + Perte_conduction_D1 + Pertes_commutation_D1 + Pertes_commutation_IGBT1);
Rendement=-3.0*Vm*Im*cosPm/(Pertes-(3.0*Vm*Im*cosPm));
y= fm;
end Variateur_pin_block;
```

Le couple résistant

```
model Couple_resistant
extends Modelica.Mechanics.Rotational.Interfaces.PartialTorque;
equation
flange.tau=1.25e-7* (der(phi)^2);
end Couple_resistant;
```

Annexe 6: Tableau des formules de calcul de l'efficacité de l'échangeur thermique en fonction du régime d'écoulement

counter flow	$\varepsilon(Z \neq 1) = \frac{1 - e^{-NTU(1-Z)}}{1 - Z e^{-NTU(1-Z)}}$ $\lim_{Z \rightarrow 1} \left[\frac{1 - e^{-NTU(1-Z)}}{1 - Z e^{-NTU(1-Z)}} \right] = \frac{1}{1 + NTU^{-1}}$ <p>Possible range: $0 \leq \varepsilon \leq 1$</p>	$NTU(Z \neq 1) = \frac{1}{Z-1} \ln \left(\frac{1-\varepsilon}{1-\varepsilon Z} \right)$ $\lim_{Z \rightarrow 1} \left[\frac{1}{Z-1} \ln \left(\frac{1-\varepsilon}{1-\varepsilon Z} \right) \right] = \frac{\varepsilon}{1-\varepsilon}$
parallel flow	$\varepsilon = \frac{1 - e^{-NTU(1+Z)}}{1+Z}$ <p>Possible range: $0 \leq \varepsilon \leq \frac{1}{1+Z}$</p>	$NTU = -\frac{\ln(-\varepsilon - \varepsilon Z + 1)}{Z+1}$
cross flow, both streams unmixed	$\varepsilon = 1 - \exp \left(\frac{e^{-NTU Z \eta} - 1}{Z \eta} \right)$ <p>with $\eta = NTU^{-0.22}$</p> <p>Possible range: $0 \leq \varepsilon \leq 1$</p>	$NTU = f(\varepsilon, NTU, Z)$ <p>must be solved numerically. However, the solution is unique (see Fig. 2).</p>
for all configurations	$\lim_{Z \rightarrow 0} \varepsilon = 1 - e^{-NTU}$	

Figure 118 : Tableau des variables des formules de calcul de l'efficacité de l'échangeur thermique en fonction du régime d'écoulement [WET 1999a]

Annexe 7: Le plug-in Simulink

Pour utiliser un composant ICAR dans Matlab/Simulink, il est nécessaire d'exploiter la notion de S-Fonction³⁰. Deux implémentations d'une S-Fonction sont possibles, en C ou en langage de script Matlab (M file). La première est plus performante en raison de son implémentation en langage compilé. La seconde, exploitant un langage interprété, est plus souple à utiliser (pas de compilation). En outre Matlab incorpore sa propre machine virtuelle Java (JVM) ce qui facilite l'utilisation des fonctionnalités Java dans le langage Matlab (sans passage par la JNI). L'implémentation en script Matlab a donc été retenue pour la réalisation du plug-in (Figure 83). Par contre, une version en C est tout à fait envisageable comme le montre le plug-in Dymola, mais cette version n'apportera pas d'avantages.

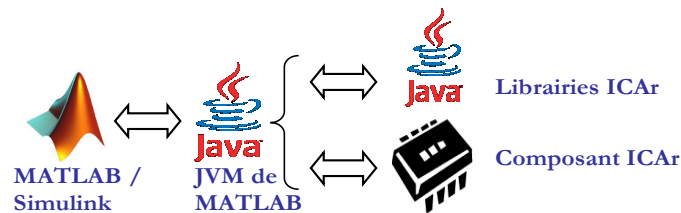


Figure 119. Le plug-in Matlab/Simulink

Le plug-in Matlab/Simulink se compose de plusieurs programmes assurant les fonctionnalités suivantes :

- Lancement de Matlab pour la prise en charge de l'ICAr: afin que Matlab reconnaisse la norme ICAR, il est nécessaire de le lancer en ayant défini les variables d'environnements nécessaires (librairies ICAR).
- Chargement et introspection de l'ICAr : un script Matlab permettant de charger et d'introspecter l'ICAr (entrées/sorties et facettes) dans l'espace de travail. Ainsi toutes les informations concernant l'ICAr, nécessaires pour l'exécution de la S-fonction correspondante, sont sauvegardées dans l'espace de travail.
- Utilisation de l'ICAr dans Simulink : Il s'agit d'un « template » de S-fonction (selon la facette) permettant l'adaptation syntaxique avec les interfaces de l'ICAr, en prenant en compte les informations d'introspection (noms des ports, ...) sauvegardées dans l'espace de travail.

En effet, un prototype d'une S-fonction de Simulink est décrit de la manière suivante [SIM 2001] :

³⁰ <http://www.mathworks.fr/help/toolbox/simulink/sfg/f6-151.html>

```
function [sys,x0,str,ts]=MySFunction(t,x,u,flag,vars, ...)
```

- T: le temps
- X : le vecteur des variables d'état
- U : le vecteur colonne des variables d'entrée (celles qui proviennent d'autres blocs simulink).
- Flag : indicateur sur les informations et calculs demandés par simulink.
- Sys: le vecteur de résultats requis par simulink, dépend de la valeur du flag

Au cours du processus de simulation, le solveur de Matlab envoie un indicateur « flag » à la S-fonction, selon les besoins de l'étape simulée. Chaque type de requête est spécifié par une valeur de la variable « flag ». Selon la valeur de cette variable, la S-fonction va exécuter une opération spécifique et fournir la sortie nécessaire « sys ».

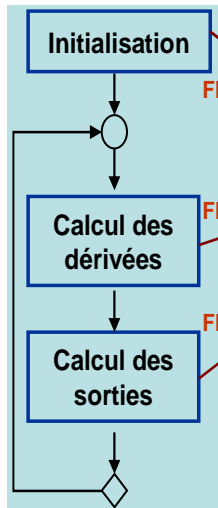
Il y a 6 types de requêtes que Simulink exécute, chacune d'elles est spécifiée par une valeur de la variable « flag » (Figure 120).

Flag	S-Function Routine	Description
0	mdlInitializesizes	Defines basic S-Function block characteristics, including sample times, initial conditions of continuous and discrete states, and the sizes array.
1	mdlDerivatives	Calculates the derivatives of the continuous state variables.
2	mdlUpdate	Updates discrete states, sample times, and major time step requirements.
3	mdlOutputs	Calculates the outputs of the S-function.
4	mdlGetTimeOfNextVarHit	Calculates the time of the next hit in absolute time. This routine is used only when you specify a variable discrete-time sample time in mdlInitializeSizes.
9	mdlTerminate	Performs any necessary end of simulation tasks.

Figure 120: Tableau des routines exécutées en fonction du flag envoyé [SIM 2010]

Par exemple, l'interaction entre le solveur de Matlab et un modèle ICAR ODE continu se fait de la manière suivante (Figure 121):

Le solveur de MATLAB



```

Flag 0 function [sys,x0,str,ts] = mdlInitializeSizes(component,stateFacet,inputNames,outputNames)
% end mdlInitializeSizes
Flag 1 function sys = mdlDerivatives(t,x,u,component,stateFacet, inputNames)
% end mdlDerivatives.
Flag 3 function sys = mdlOutputs(t,x,u,component,stateFacet, inputNames, outputNames)
% end mdlOutputs
  
```

Le composant Logiciel

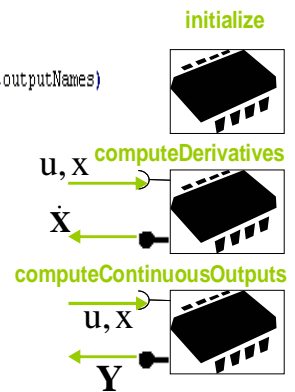


Figure 121. Illustration de l'interaction entre le solveur de Matlab/la S-fonction et l'ICAr

Annexe 8: Le plug-in DYMOLA

La réalisation du plug-in du composant ICAR vers Dymola se décompose ainsi en trois principales parties

- L'interfaçage entre les langages Java et C via la JNI.
- L'interface entre les langages C et Modelica via la fonctionnalité « external ».
- La génération automatique du modèle Modelica correspondant à l'ICAr.

L'interfaçage entre le composant ICAR en Java et le programme en C

Le composant logiciel ICAR est développé en Java pour bénéficier d'une portabilité quelque soit l'environnement de travail et d'une architecture évolutive. Son utilisation dans le langage C nécessite le passage par la JVM (Java Virtual Machine) pour l'exécution de ses différentes interfaces Java (Figure 122). Cette méthode est donc accompagnée par une phase d'interfaçage permettant au code java exécuté à l'intérieur de la machine virtuelle d'être appelé par le langage natif C au moyen de la JNI (Java Native Interface). La JNI permet à une application Java d'interagir avec une application native (C, fortran, ...). Elle spécifie le « mapping » des types entre les deux langages et permet la création des objets Java et l'appel de ces méthodes.

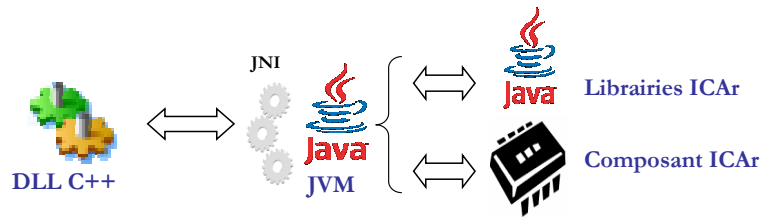


Figure 122. L'interfaçage ICAR / C++

Un projet DLL est construit pour rassembler les différentes fonctions C permettant d'effectuer les manipulations de lancement de la JVM et d'interaction avec le composant ICAR.

L'interfaçage entre Modelica et le programme C

Il est possible d'appeler une méthode C dans Dymola en y décrivant une fonction correspondante. Cette fonction Modelica permet de déclarer les entrées/sorties adaptées à cette fonction externe, spécifier le langage dans laquelle elle est décrite (« C ») ainsi que son nom (Figure 123). Les fichiers nécessaires à cet appel peuvent être spécifiés dans les annotations grâce aux mots clés « include » et « library » (pour les librairies) [SPEC 2010].

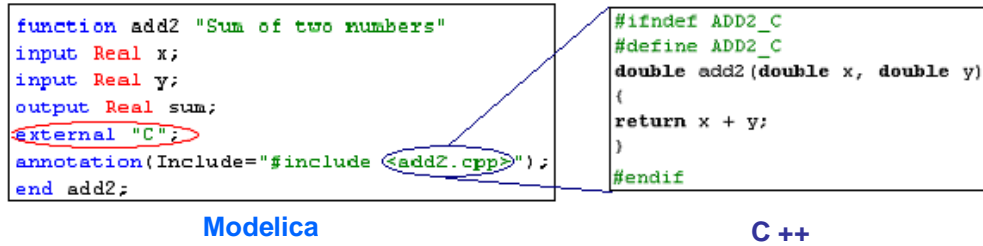


Figure 123. Exemple d'appel de fonction externe C dans Modelica

Un package de fonctions Modelica a donc été défini pour permettre l'appel de l'ensemble des fonctions externes décrites dans la DLL d'interfaçage avec le composant ICAr.

La génération automatique du modèle Modelica

Un modèle Modelica doit déclarer ses variables et paramètres avant de les utiliser au sein de ses équations ou/et algorithmes. Cette partie du modèle est spécifique à chaque composant ce que nécessite une introspection préalable en vue d'en créer le modèle Modelica correspondant. Cette procédure est automatisée par un programme Java à l'issue duquel, un fichier Modelica spécifique (.mo) est généré.

Ce modèle contient en plus de la déclaration de ses variables et paramètres, le package de définition des fonctions externes et la séquence de leurs appels (Figure 124). Ces fonctions externes vont à leur tour appeler les interfaces correspondantes de la facette de l'ICAr implémentée [GAA 2011]. Les fonctions utilisées ainsi que leur ordre d'appel (séquence) dépendent de la nature du modèle (continu, hybride...) et ainsi du type de la facette implémentée dans l'ICAr.

```

model ICArCOMFIEODE
  parameter String path= "../ICArCOMFIE.icar";
  parameter Real P1=3500;
  parameter Real P2=0;
  Real[2] outputs;
  String[2] outputsNames={"T1","T2"};
  Real[16] stateVariables;
  String[16] stateVariablesNames={"X1","X2","X3","X4","X5","X6","X7","X8","X9","X10","X11","X12","X13","X14","X15","X16"};
protected
  constant Integer icar_stateVariableNumber=16;
  Package des fonctions externes
initial equation
  IcarFunctions.Initialize(path);
  stateVariables=IcarFunctions.initialStateVariables_Retrieve(icar_stateVariableNumber);
  IcarFunctions.setScalarInputs({"P1","P2"},{P1,P2});
equation
  IcarFunctions.setStateVariableInputs(stateVariablesNames,stateVariables);
  (outputs)=IcarFunctions.scalarOutputsComp_Retrieve(time,outputsNames);
  (der{stateVariables})=IcarFunctions.stateVariableDerivatives_Retrieve(time,icar_stateVariableNumber);
end ICArCOMFIEODE;

```

Figure 124. Exemple du code Modelica automatiquement généré (ICAr ODE de COMFIE)

Annexe 9: Le plug-out COMFIE

Le logiciel COMFIE n'offre actuellement pas d'API pour l'interaction avec ses modèles ou son cœur de calcul. Le lancement d'une simulation COMFIE, après avoir décrit le bâtiment étudié (enveloppe, météo, scénarios..), permet de traiter l'enveloppe du bâtiment et d'établir le système d'état qui correspond à sa modélisation (chapitre 2). En lien avec les développeurs du logiciel, nous avons obtenu une fonctionnalité d'export de ce système d'état ainsi que de la matrice des sollicitations, dans des fichiers texte [GAA 2011] (Figure 125).

Le programme principal du plug-out de COMFIE, développé en Java, permet de récupérer les fichiers contenant les différentes matrices, les analyser afin de définir les entrées sorties des modèles et en extraire les différentes matrices afin de reconstruire le système d'état décrivant le modèle en complétant les différentes interfaces de l'ICAr. Chaque interface du composant va assurer l'exécution d'une opération de calcul du système d'état global [GAA 2012].

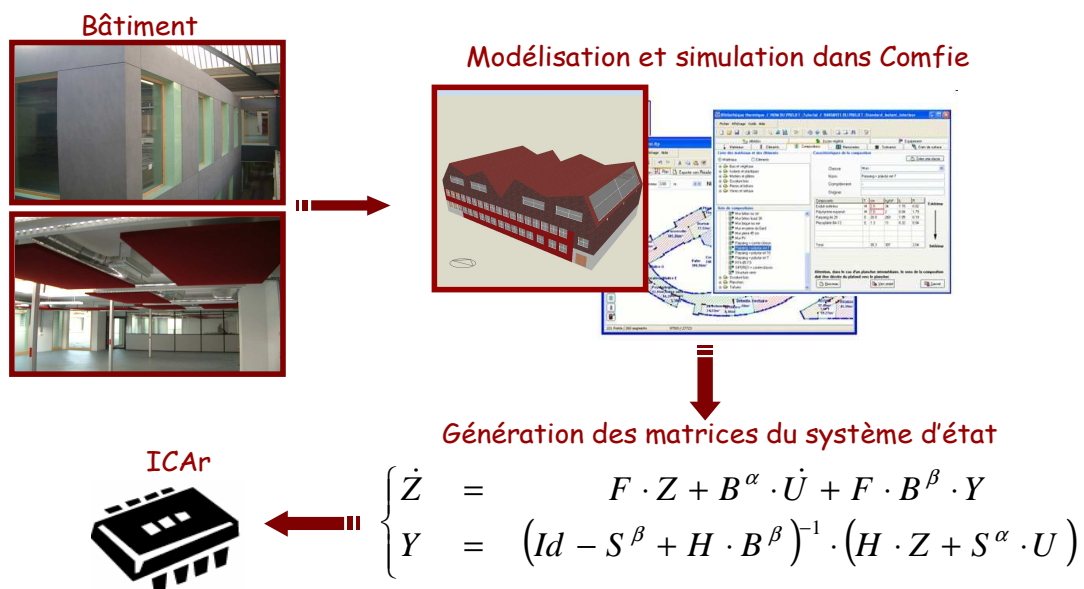


Figure 125. Le principe du plug-out d'un modèle de COMFIE

Avec $Z = X - B^\beta \cdot Y$ et Id la matrice identité

Plusieurs facettes du composant peuvent être implémentées selon les objectifs d'utilisation du composant (facette ODE continue et discrète, et facette de co-simulation). Plusieurs types du système d'états peuvent être reconstruits pour alimenter ces interfaces (équations 2 et 3 du chapitre 2).

La figure suivante (Figure 102) illustre un exemple d'implémentation d'ICAr ODE continu à partir du système d'état résultant du traitement de l'enveloppe dans COMFIE.

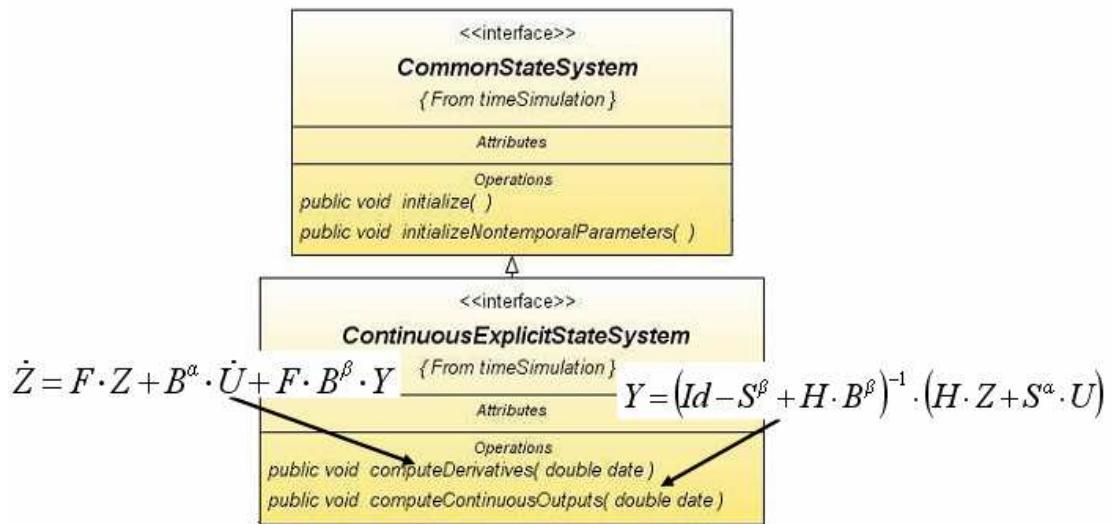


Figure 126. Implémentation de l'interface ODE de l'ICAr COMFIE

Annexe 10: Le plug-in TRNSYS

La réalisation du plug-in TRNSys se base sur les deux autres plug-in (Dymola et Matlab) développés précédemment. Il reprend la partie d'interfaçage entre les langages C++ et Java (en passant par la JNI) et réutilise la même DLL construite pour le plug-in Dymola. D'un autre côté, TRNSys, comme Matlab/Simulink avec les S-fonctions, offre une interface pour l'introduction des nouveaux modèles : les « Type ».

En effet, l'introduction d'un nouveau modèle dans TRNSys est possible via la création d'un nouveau « TYPE » en C, C++ ou en fortran [TRN 2007]. Le langage C++ été choisi afin de pouvoir profiter des développements réalisés. Un « template » (proforma) correspondant à ce modèle est généré automatiquement par TRNSys (Figure 127) :

```
int TYPE500 (
    double &time, // the simulation time
    double xin[], // the array containing the component INPUTS
    double xout[], // the array which the component fills with its appropriate OUTPUTS
    double &t, // the array containing the dependent variables
    double &dt, // the array containing the derivatives of T which are evaluated
    double par[], // the array containing the PARAMETERS of the component
    int info[], // the information array
    int icntrl // the control array
)
```

Figure 127. Le prototype d'un nouveau « TYPE » de TRNSys

Le tableau *info* dans le *Type* (Figure 127) joue le rôle du *flag* de la S-fonction, comme intermédiaire entre le solveur et le modèle. Selon la requête renvoyée par le solveur (correspondante à une étape de simulation donnée), une fonction C du « type » sera exécutée.

Le type « info » communique beaucoup plus d'informations au modèle que le « flag » de Simulink (Figure 128):

INFO(n)	Role
1	Unit number
2	Type number
3	Number of Inputs in the input file (.dck) for this Unit
4	Number of Parameters in the input file (.dck) for this Unit
5	Number of Derivatives in the input file (.dck) for this Unit
6	Number of outputs (set by the Type)
7	Number of iterative calls to this Unit in the current time step. Special values: -1, -2
8	Total number of calls to this unit. Special values: -1
9	Indicate whether or not the Type depends on the passage of time and tells TRNSYS where the Type should go in the calling sequence
10	Used in storage management for TRNSYS 15 Types (not used by TRNSYS 16 Types)
11	Indicates number of discrete control variables
12	TRNSYS version for which the Type was written (15 or 16)
13	Indicates when all Units have converged in the current time step
14	Reserved for future use
15	Reserved for future use

Figure 128. Tableau des informations communiquées par les éléments du vecteur « info »

La valeur de info(7) permet de déterminer l'étape de la simulation encours de traitement (Figure 129).

Table 8.4.3.3-1: INFO(7) values

INFO(7)	Role
-2	Special call at the very beginning of the simulation to allow for Type version signing. Types should only set the value of INFO(13) during this call
-1	Initial call in simulation for this Unit. The Type should only perform initialization operations here: array sizing, memory allocation, file opening, parameter checking, etc. The Type should also set the value of INFO(6), INFO(9) and INFO(10) (if used)
0	First call in the current time step for this Unit.
1	First iterative call (second call) in the current time step for this Unit
2	Second iterative call (third call) in the current time step for this Unit
+n	n th iterative call ((n+1) th call) in the current time step for this Unit

Figure 129. Tableau des informations communiquées par la valeur de l'élément 7 du vecteur « info »

Les fonctions C vont appeler à leur tour une interface Java de l'ICAr qui va s'exécuter dans une JVM. La JNI assurera l'adaptation des données entre la JVM et le langage C (Figure 130).

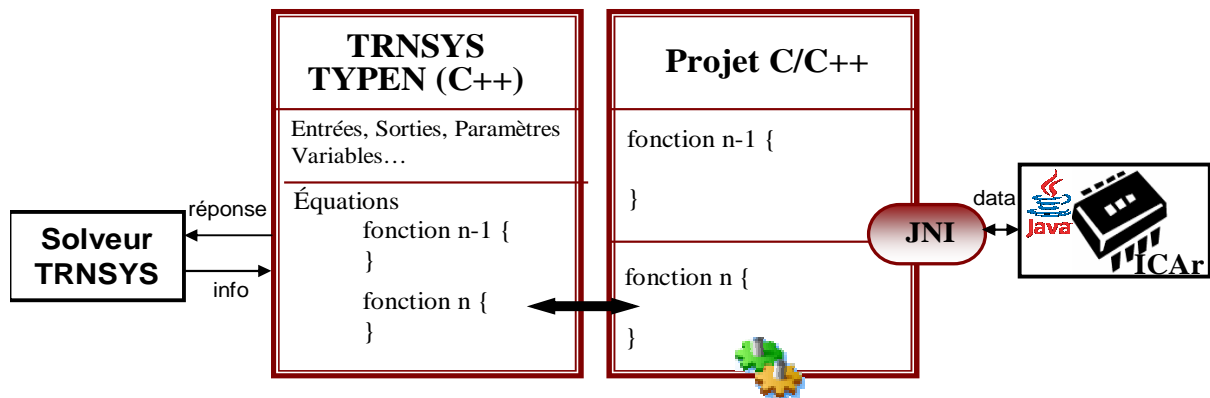


Figure 130. Le principe du plug-in TRNSYS

Bien que la partie du plug-in concernant les équations relatives au Type « C » de TRNSYS soit identique pour tout composant logiciel implémentant la même facette, la définition des variables du modèles (entrées, sorties, paramètres...) dépend du modèle et nécessite un traitement au cas par cas. Dans l'état actuel, cette phase se fait manuellement mais une automatisation est envisageable dans le cadre d'une collaboration avec le CSTB (projet Plume). Dans la perspective de ce travail, un plug-out d'un composant logiciel depuis la DLL d'un type de TRNSYS sera également étudiée.

Résumé

Pour mieux maîtriser ses flux énergétiques et respecter les diverses restrictions mises en place dans ce secteur énergivore, le bâtiment devient un système de plus en plus complexe incluant divers technologies innovantes comme les systèmes de gestion énergétiques (SGEB), une isolation performante et intégrant les énergies renouvelables. Cette complexité exige un changement dans les techniques et paradigmes actuels de simulation du bâtiment pour la prise en compte de ses diverses évolutions. Une modélisation globale des différents composants de ce système et une simulation efficace de ses sous-systèmes hétérogènes doivent être dorénavant assurées.

Ces objectifs ne pourront être atteints qu'à travers l'exploitation des approches méthodologiques d'interopérabilité. Plusieurs solutions d'interopérabilités ont été exploitées dans le secteur du bâtiment. L'état de l'art dans ce secteur, met l'accent sur le manque de standardisation des solutions appliquées. Une approche boîte blanche se basant sur le langage Modelica a remarquablement émergée. Pour monter ses intérêts ainsi que ses limites, cette solution est adoptée pour la modélisation du système de bâtiment «PREDIS», à haute performance énergétique. Une approche boîte noire complémentaire, s'appuyant sur le standard de composant logiciel dédié à la simulation, est également mise en oeuvre pour palier aux difficultés rencontrées en utilisant la première approche de modélisation système. Cette approche s'articule autour du concept de bus à composants permettant une interopérabilité effective entre outils de modélisation et environnements de simulation.

En plus de l'architecture logicielle autour de la plateforme d'interopérabilité, une simulation efficace du système hétérogène requière des techniques de simulations adaptées. Ces dernières peuvent exiger des adaptations des modèles utilisés qui sont prévues par la norme de composant.

Mots clés: Efficacité énergétique, modélisation, simulation système, boîte blanche/boîte noire, Modelica, bus à composants logiciels, plug-in, plug-out.

Abstract

To better reduce its invoices, control its energy flows and respect various restrictions in this sector characterised by important consumption, the building becomes more and more complex including various innovative technologies such as Energy Management Systems (BEMS), efficient insulation and integrating renewable energies. This complexity requires a changing in building simulation techniques and paradigms in order to take into account its various developments. A global modelling of this system taking into account its various components and ensuring an efficient simulation of its heterogeneous subsystems must be performed.

These objectives can only be achieved through the use of interoperability methodological approaches. Several interoperability solutions have been explored in the building sector and the state of the art make an accent on the standardization lack of applied solutions. A white box approach based on Modelica language has emerged in this area. To raise its interest and limitations, this solution is adopted for "PREDIS" system, a high energy performance building, modelling. A complementary black box approach, based on software component standard and dedicated for simulation is also applied to overcome the first approach difficulties. This approach is based on software component bus concept that is able to ensure an effective interoperability between modelling tools and simulation environments.

In addition of the established software architecture around the platform interoperability, an efficient simulation of heterogeneous systems requires appropriate simulations techniques. These techniques may require several adaptations of used models that are provided by the component standard.

Keywords: Energy efficiency, modelling, system simulation, white box/black box, Modelica, software component bus, plug-in, plug-out.