



HAL
open science

Génie logiciel et interopérabilité entre modèles et outils

Benoît Delinchant

► **To cite this version:**

Benoît Delinchant. Génie logiciel et interopérabilité entre modèles et outils. Livre blanc sur les recherches en énergétique des bâtiments, Groupe d'Analyse Prospective Thématique Bâtiment et Ville Durables, Presse des Mines, pp.183 à 191, 2013. hal-00997110

HAL Id: hal-00997110

<https://hal.science/hal-00997110>

Submitted on 21 Jan 2019

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.

3.4.3 GENIE LOGICIEL ET INTEROPERABILITE ENTRE MODELES ET OUTILS

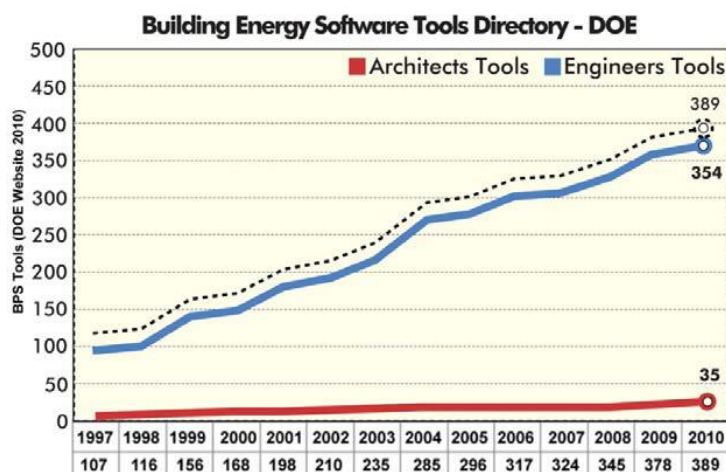
B. DELINCHANT

Le besoin d'interopérabilité

L'interopérabilité entre les modèles et les outils que nous utilisons est aujourd'hui une nécessité. Le génie logiciel nous apporte un certain nombre de techniques qu'il convient d'investiguer en vue de les appliquer et/ou adapter à notre domaine qui est celui de la modélisation, de la simulation, de la conception, de la gestion embarquée, etc. en énergétique du bâtiment.

Pourquoi l'interopérabilité est-elle nécessaire ? Il faut tout d'abord considérer la quantité importante d'outils et de méthodes de modélisation disponibles. 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, aux énergies renouvelables et au développement durable dans le secteur du bâtiment. Le nombre de ces outils a 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 ci-dessous).

Cette remarquable évolution est due à une conscience des enjeux majeurs de ce secteur [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... Cette tendance ne semble pas s'infléchir puisque aujourd'hui ce sont des quartiers complets qui doivent pouvoir être simulés intégrant des phénomènes couplés de plus en plus nombreux.



Les outils de simulation du bâtiment développés entre 1997 et 2010 [ATT 2011]

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.

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.

Il faut considérer également l'ensemble du cycle de vie du bâtiment dans lequel la simulation intervient. Or, chaque stade du cycle de vie apporte des contraintes qui lui sont propres, faisant qu'il est envisageable de considérer des modèles différents pour chaque objectif. En effet, les contraintes en conception (ou phase d'esquisse) font qu'il n'est pas judicieux de disposer de modèles trop détaillés [WUR 2012]. De la même manière qu'en gestion, durant la phase d'exploitation du bâtiment, il est commode de disposer de modèles rapides permettant d'anticiper sur les quelques prochaines heures, ses modèles devant également pouvoir se connecter facilement à des techniques d'optimisation [MISS 2012].

Tout ceci fait que les outils ont eux aussi un cycle de vie, et qu'il est donc utopique de chercher à réaliser un unique outil, capable de traiter tous les objectifs dans lesquels les modèles de simulation sont exploités. Ainsi, nous considérons l'interopérabilité entre les modèles et les outils comme la solution incontournable pour parvenir à gérer la complexité du système bâtiment.

L'interopérabilité permet ainsi de réaliser par exemple des co-simulations [TRCK 2006], dont les principaux intérêts sont :

- Bénéficier des compétences et performances spécialisées de chaque outil.
- Permettre des couplages hétérogènes en termes de méthode de discrétisation des équations aux dérivées partielles (PDE) (mono zone, éléments finis...), de formalismes de simulation (équations différentielles ordinaires (ODE) ou algébriques (DAE), à temps discret ou modèle à événements), ou de méthodes de résolution (intégration à pas fixe/variable ...).
- Prototyper facilement de nouveaux systèmes par cette approche modulaire dans laquelle un outil peut être ajouté ou substitué par un autre.

Etat des lieux de l'interopérabilité en simulation du bâtiment

Selon la nature des modèles, la solution de couplage mise en œuvre pour atteindre l'interopérabilité sera différente. Nous distinguons différentes natures selon une formulation du modèle sous forme explicite et ouverte, par un code binaire fermé, ou un compromis entre ces deux approches.

D'un côté, les modèles peuvent être décrits dans un formalisme descriptif, pour lequel les équations, fonctions, ou algorithmes, sont explicitées ; dans ce cas on parlera de « boîte blanche » ou de « boule blanche ». C'est par exemple le cas pour des modèles décrits dans des langages d'environnements génériques (par exemple : Matlab ou Scilab), ou encore dans des langages normalisés pour la simulation dynamique (par exemple : Modelica ou VHDL-AMS). Les terminologies « boule » et « boîte » sont reliées à la causalité, c'est-à-dire à l'orientation de leurs ports. Une boîte possèdera des entrées/sorties, tandis qu'une boule disposera de ports non-orientés exploitables dans une composition (ex : la différence de potentiel et le flux d'une résistance).

D'un autre côté, les modèles peuvent être accessibles sous forme binaire. Généralement, les composants ainsi modélisés apparaissent dans les bibliothèques de modèles d'environnements de simulation (TRNSys, Energy+, etc.). L'utilisateur accède à un élément graphique qu'il dépose sur une feuille, et qu'il va connecter à d'autres éléments au travers de ses ports. Une aide permet de comprendre les hypothèses et le paramétrage du modèle, mais ses équations sont généralement rendues inaccessibles, on parle alors de boîtes noires.

Ainsi, l'interopérabilité entre modèles peut être examinée selon ces deux directions. La première (boîte/boule blanche) nécessite la définition de langages standards telles que Modelica. Par contre, un tel langage ne peut se prétendre exhaustif et vouloir traiter tous les formalismes de modélisation, c'est

par exemple le cas des équations aux dérivées partielles (PDE) dans Modelica [MAZ 2009][LI 2009]. Une approche d'interopérabilité est donc nécessaire même dans ces formalismes descriptifs. Il n'existe pas réellement aujourd'hui de solution permettant l'interopérabilité boîte blanche de langages hétérogènes, ce qui nécessiterait le développement d'un outil d'interprétation de N langages. Dans les faits, les modèles sont souvent retranscrits dans un unique langage ce qui nécessite des adaptations syntaxiques, voire parfois des adaptations sémantiques conduisant à des modifications des hypothèses de modélisation.

La deuxième direction, complémentaire à la première, traite des modèles « boîte noire ». La réutilisation de tels modèles est plus aisée car ils n'exposent qu'une interface généralement définie par les variables et paramètres du modèle. Ainsi, le traitement mathématique permettant de calculer des grandeurs de sortie à partir de grandeurs d'entrée peut être quelconque. Une interopérabilité effective dans ce domaine n'est aujourd'hui pas atteinte, seuls des travaux de recherche ont été réalisés pointant les difficultés pour sa mise en œuvre [RAD 2005]. Des cas de co-simulations établis dans le bâtiment peuvent être répertoriés selon les deux typologies: Une co-simulation directe entre deux outils [JAN 1999][DJU 2003][KEI 2002], ou une co-simulation orchestrée par un environnement de co-simulation [WET 2008][SAG 2011][GU 2011]. Il s'agit ici de cas de co-simulation, faisant donc interopérer des modèles via des outils de modélisation/simulation. Ces couplages entre outils correspondent à des développements réalisés au cas par cas.

Outre la co-simulation, interopérabilité la plus classique, le couplage de modèles offre un compromis intéressant entre le couplage fort (boîte blanche) et le couplage faible par co-simulation. Il ne s'agit plus de faire dialoguer des logiciels possédant leur propres solveurs, mais de composer un système de modèles offrant une sémantique compatible, le tout connecté à un solveur. Pour réaliser ce couplage, les modèles doivent expliciter par exemple, dans le cas de la simulation dynamique de type ODE, les dérivées temporelles des variables d'état en fonction de leurs valeurs et des sources. Si plusieurs modèles respectent cette sémantique, ils peuvent être couplés, via une adaptation syntaxique probable. C'est par exemple le cas des « types » TRNSys, couplés à d'autres modèles dans l'environnement « Simulink » [RID 2009].

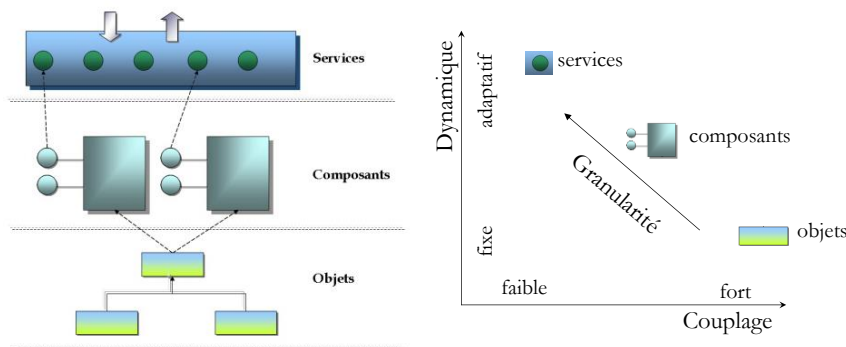
Limites de l'approche et perspectives de recherche

L'interopérabilité est incontournable mais les solutions de couplage permettant à un bureau d'étude d'étudier rapidement une idée innovante de conception, ou à un exploitant gérant des bâtiments d'intégrer des solutions intelligentes et efficaces à moindre coût, ne sont pas disponibles. L'état de l'art nous montre que les bénéfices de l'interopérabilité sont grands mais que sa mise en œuvre demande des efforts considérables. Il est donc impératif de travailler dans la direction d'une plus grande interopérabilité entre les modèles et outils dont on dispose actuellement mais aussi pour les prochaines générations.

Pour y parvenir, la solution la plus raisonnable est d'exploiter les outils que le génie logiciel nous offre. En effet, l'interopérabilité est en premier lieu un problème informatique et doit être traité en tant que tel. Par contre, notre rôle est d'adapter ces solutions à nos problématiques de co-simulation et couplage de modèles avec les contraintes spécifiques qu'elles apportent.

Vis-à-vis de la problématique de couplage boîte noire, le génie logiciel nous apporte la notion d'interface. Ces interfaces apparaissent dans les concepts successivement inventés de programmation orientée objet, programmation orientée composant et programmation orientée service [Don 2006]. Comme l'illustre la figure suivante, une structure hiérarchique à 3 niveaux de granularité peut être définie en termes de réutilisation. Le concept de services agrège celui de composants, agrégeant lui-même celui d'objets, chaque couche offrant ses propriétés en termes de compromis de

dynamique et de force de couplage. Plus le couplage est faible plus la dynamique est forte, c'est-à-dire que les couplages peuvent être modifiés plus rapidement et facilement.



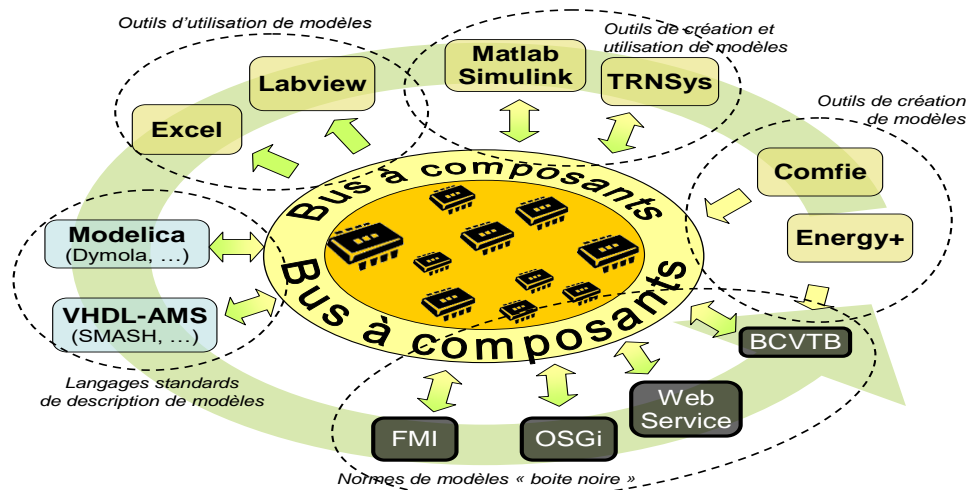
Architecture hiérarchique à objets/composant/services, et compromis dynamique / couplage

La programmation orientée objet est généralement appliquée au sein d'un outil, nous n'en parlerons donc pas ici. La programmation orientée composant offre par contre une solution très intéressante au niveau de la réutilisation de modèles hétérogènes pour de la composition de systèmes. La programmation orientée service est quant à elle bien adaptée à la réutilisation des outils pour de la co-simulation par exemple.

Concernant les composants, des efforts considérables ont été effectués dans le secteur automobile à travers le projet européen Modelisar, conduisant au développement de la norme FMI¹ et permettant aux différents partenaires d'échanger des modèles dédiés à la simulation dynamique. Dans le secteur du bâtiment, la norme ICAr [GAA 2011] a été mise au point dans plusieurs travaux et dans plusieurs projets de recherche² pour aboutir à la notion de bus à composants (voir figure ci-dessous). Ce dernier doit être architecturé autour d'une norme de composant logiciel et offrir des passerelles génériques à partir des outils de modélisation et vers les outils de simulation du domaine. Ce bus doit également intégrer d'autres normes pour garantir une interopérabilité maximale. Ces normes sont celles des standards de langage de description de modèle (Modelica ou VHDL-AMS) mais aussi d'autres normes « boîte noire » offrant par exemple les capacités de modélisation d'Energy+ via la norme de modèle définie dans BCVTB [WET 2008].

¹ <https://www.fmi-standard.org/>

² ANR SimintheC, ANR Plumes, ANR Fiabilité, Climb



Le concept de BUS architecturé autour d'une norme de composant logiciel représentant des modèles.

En comparaison à l'approche orientée services, la programmation orientée composant offre la capacité de déploiement du modèle chez ses différents utilisateurs. Pour ce faire, les modèles doivent embarquer dans le composant toutes ses dépendances le rendant ainsi autonome. Lorsque cette autonomie n'est pas possible, ou trop complexe à obtenir, l'approche orientée service prend alors le relais. La différence principale dans ce cas réside dans le fait que l'exécution du modèle est délocalisée sur le « cloud³ ». Les modèles ne sont ainsi plus échangés, seule leur adresse « URL » est fournie, le modèle et toutes ses dépendances (logiciel de modélisation, licence, ...) ayant été déployées préalablement sur un serveur [DEL 2012]. Cette solution bénéficie de nombreux avantages dans le cas de simulations nécessitant des ressources de calcul importantes, le partage de licences logicielles, etc. [END 2004].

Outre la notion d'interfaçage d'un modèle ou d'un outil que nous venons d'aborder et qui représentent l'approche d'interopérabilité la plus généralisable (boîte noire), il convient également d'aborder l'interopérabilité de modèles descriptifs (boîte/boule blanche). Comme nous l'avons souligné précédemment, cette interopérabilité se traduit aujourd'hui concrètement par la retranscription souvent manuelle des modèles, d'un langage à un autre. L'informatique, là encore, peut nous offrir des outils, ceux relatifs à l'ingénierie dirigée par les modèles (IDM ou MDE en anglais) [SCH 2006]. Le concept clé, porté par cette discipline, est celui de positionner le modèle descriptif indépendamment de tout environnement visant à son exécution. L'application dans notre domaine se traduit donc par la séparation entre modélisation et simulation. l'IDM propose alors une architecture orientée modèle (MDA), dans laquelle apparaissent divers niveaux d'abstraction telle que la meta-modélisation définissant le langage de description d'un modèle. Ce niveau de méta-modélisation offre le cadre formel permettant de développer des projeteurs, réalisant la traduction automatique d'un langage vers un autre. En particulier, on utilise ces projeteurs pour traduire un modèle indépendant d'une plateforme d'exécution (PIM) vers un ou plusieurs modèles spécifiques à différentes plateformes d'exécution (PSM). Pour être concret, il peut par exemple s'agir d'un modèle Modelica (PIM) qui serait traduit (projeté) vers le langage d'un environnement de simulation dynamique ou vers celui d'un algorithme d'optimisation [VER 2012].

³ Cloud computing : concept qui consiste à déporter sur des serveurs distants des stockages et des traitements informatiques traditionnellement localisés sur des serveurs locaux ou sur le poste de l'utilisateur

Nous voyons donc qu'il existe un certain nombre de pistes offertes par le génie logiciel pour nous aider à atteindre l'interopérabilité nécessaire entre nos modèles et outils de modélisation. Enfin, pour conclure ces perspectives de recherche, nous n'avons pu aborder ici que l'interopérabilité entre modèles comportementaux, mais l'interopérabilité entre les données du bâtiment et ces modèles, représente également un enjeu majeur [STE 2012].

Références bibliographiques

- [AUG 2002] G. Augenbroe, « Trends in buildings simulation », Building and environment Journal, N° 37, pp 891-902, 2002.
- [ATT 2010] S. Attia, « Building Performance Simulation Tools: Selection Criteria and User Survey », rapport de recherche, 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.
- [DEEU 2011] Statistiques du département de l'énergie des états unis, disponible sur <http://apps1.eere.energy.gov/buildings/tools_directory/>, 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.
- [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.
- [DEL 2012] B. Delinchant, P.Y. Gibello, F. Verdière, F. Wurtz « Distribution des services de calcul pour la conception et la gestion optimale des bâtiments: perspectives des approches composants déployés via le 'cloud computing' », Rencontres AUGC/IBPSA France, Chambéry, 06-08 juin 2012
- [DON 2006] Didier DONSEZ « Objets, composants et services : intégration de propriétés non fonctionnelles », HDR Informatique, 11/12/2006
- [END 2004] M. Endrei and al, "Patterns: Service-oriented architecture and web services", <http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf> , 2004
- [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.
- [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.
- [JAN 1999] M. Janak, "Coupling building energy and lighting simulation", Proceedings of the 5th International IBPSA Conference, Kyoto, Japan, 1999.
- [KEI 2002] W. Keilholz, « TRNSYS World-wide », IBPSA news, volume 12, numéro 1, 2002.
- [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.
- [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.
- [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.
- [MORA 2009] L. Mora, « Etat de l'art en termes de modèles et d'outils de simulation », livrable Projet SIMBIO, 2009.

- [RAD 2005] M. Radosevic, J. Hensen, and A. Wijsman, « Implementation strategies for distributed modeling and simulation of building systems », IBPSA conference, Montréal, Canada, August 2005
- [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.
- [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.
- [SCH 2006] Schmidt, D.C. « Model-Driven Engineering » , February 2006, IEEE Computer 39 (2).
- [STE 2012] Model interoperability in building information modelling, Jim Steel, Robin Drogemuller, Bianca Toth, Software System Modelling (2012) 11:99–109
- [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.
- [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, 28 - 30 March 2012, Cambridge, UK. pp.340-344
- [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 2011] M. Wetter, « A view on future building system modeling and simulation », in "Building Performance Simulation for Design and Operation", 2011, Jan L. M. Hensen and Roberto Lamberts, Routledge, UK, ISBN: 978-0-415-47414-6
- [WUR 2012] F. Wurtz, X. Brunotte, W. Basset, S. Ploix, R. Marten, J. Pouget, Y. Riffonneau « Dimensionnement optimal et simultané de l'enveloppe, des systèmes et de la stratégie de gestion en phase d'esquisse : application aux gares à énergie positive », Rencontres AUGC/IBPSA France, Chambéry, 06-08 juin 2012