

MODELISATION, SIMULATION ET SYSTEME EXPERT EN THERMIQUE DU BATIMENT (2ème partie et fin)

Par

Sidi Mohamed Karim EL HASSAR

Département Technologie et Structures des Ouvrages

Centre National d'Etudes et de Recherches Intégrées du Bâtiment, Algérie

Roger PELLETRET

Service Informatique et Bâtiment

Centre Scientifique et Technique du Bâtiment, France

Résumé

Nous présentons ici quelques options existantes pour réaliser un environnement de simulation dans le domaine de la thermique du bâtiment. Pour ce faire, la méthodologie adoptée a consisté à développer tout d'abord le concept d'ESI : Environnement de Simulation Intelligent ; ce concept qui repose sur un modèle abstrait d'environnement de simulation doit permettre la réalisation d'environnements de simulation génériques. La réalisation d'un ESI est un travail complexe qui fait appel à des connaissances issues de plusieurs domaines : modélisation physique, analyse numérique, génie logiciel, intelligence artificielle.

Quelques méthodes numériques ont été présentées dans la 1ère partie de l'article : les méthodes aux différences finies qui occupent une place importante dans le domaine de la thermique du bâtiment, les méthodes modales, méthodes d'avenir, dont l'application très récente au domaine du bâtiment n'a pas encore permis d'en exploiter toutes les ressources.

Nous présentons dans cet article quelques points qui permettent de concevoir une interface graphique. L'interface ne doit être conçue qu'après avoir pris connaissance des besoins et des habitudes de travail des utilisateurs ; elle doit être réalisée autour des idées suivantes : multi-fenêtrage, manipulation directe, archivage aisé des projets, aide "experte" intégrée au logiciel. L'aide "experte" peut être conçue autour de bases de données et de systèmes experts ; elle peut aussi faire appel à des systèmes experts fondés sur le raisonnement par cas. Il ressort de cette réflexion qu'une interface graphique "bien" conçue permet d'atteindre la plupart des objectifs visés par la concrétisation du concept d'ESI.

Mots clés : modélisation - simulation - bâtiment - environnement de simulation intelligent - dialogue Homme/Machine - système expert - programmation orientée objet.

1 CONTEXTE

En Algérie, les professionnels du bâtiment ne tiennent pas suffisamment compte des phénomènes thermiques au moment de la conception et au moment de la réalisation de leurs projets ; ainsi, de fortes consommations d'énergie pour le chauffage et la climatisation sont nécessaires pour répondre aux exigences de confort.

La mise au point d'environnements de simulation dans le domaine de la thermique du bâtiment va faciliter le travail des professionnels en les aidant à résoudre certains problèmes tels que :

- ✓ Comment concevoir une habitation thermiquement isolée d'une façon optimale, c'est-à-dire qui répond aux exigences de confort à un coût raisonnable ?
- ✓ Comment vérifier qu'une construction est conforme à la réglementation thermique ⁽¹⁾ ?
- ✓ Comment faire en sorte qu'une habitation déjà construite devienne conforme à la réglementation thermique ?

2 INTRODUCTION

Cette publication fait référence à des travaux de recherche

⁽¹⁾ La réglementation thermique algérienne est en cours d'élaboration dans un projet Maghreb-CEE.

ches menés dans le cadre du développement par le Centre Scientifique et Technique du Bâtiment (France) d'un Environnement de Simulation Intelligent (ESI) destiné à faciliter l'accès aux logiciels de calculs scientifiques dans le secteur bâtiment [7]. En particulier, deux travaux de recherches complémentaires ont contribué à l'évolution du concept d'ESI :

- ✓ le premier travail [8] s'est attaché à définir le concept d'ESI et à développer un modèle profond d'un tel environnement, modèle générique dont le but est de faciliter la réalisation d'applications dédiées ;
- ✓ le second travail [3], tout en contribuant à la définition des spécifications de l'environnement de simulation générique, a essentiellement porté sur la réalisation d'une application concrète que l'on peut considérer comme une phase expérimentale de mise en œuvre des concepts développés ; pour ce faire, un environnement fondé sur le principe de la modélisation par assemblage a été réalisé, cet environnement est appelé IISIBât (Interfaces Intelligentes pour la Simulation dans le Bâtiment) ; l'application a été réalisée autour du logiciel de calculs thermiques TRNSYS ⁽²⁾ ; en outre, ce travail de recherche a permis de définir pour de tels environnements les possibilités d'utilisation des systèmes à bases de connaissances (buts et moyens, évaluation de la pertinence des solutions).

L'interface IISIBât a été réalisée avec le langage Le-Lisp ⁽³⁾, enrichi d'une extension capable de gérer des objets graphiques : la boîte à outils Aïda ⁽⁴⁾ :

- Le-Lisp est un descendant du langage LISP, qui fut développé au laboratoire d'Intelligence Artificielle du Massachusetts of Technology (MIT). Celui-ci donna lieu à la production de nombreux dialectes, Le-Lisp s'en distinguant par le fait qu'il s'appuie sur une machine virtuelle indépendante du système d'exploitation. De plus, Le-Lisp (version 15-25) est muni d'une extension objet (Ceyx), permettant une programmation orientée objet. Le-Lisp fournit des mécanismes puissants pour appeler des programmes écrits en d'autres langages (comme C, Fortran ou Pascal) ; cela permet de séparer de façon claire le programme (écrit en Le-Lisp, en C, ou en Fortran) de l'interface graphique construite à partir de l'environnement Aïda ;
- La boîte à outils Aïda est liée à la bibliothèque X-Window ; elle offre la possibilité de programmer des interfaces graphiques à partir de composants élémentaires, en utilisant les techniques d'héritage. Elle se compose d'une bibliothèque de plus de 100 classes de composants graphiques prédéfinis simples (boutons poussoirs, boutons radio, cases à cocher) ou complexes (barres de défilement, éditeurs d'arbres, éditeurs de texte, éditeurs de courbes). L'utilisation de la boîte à outils Aïda (version 1.65) consiste à définir des structures, sous-structures des composants graphiques cités plus haut. La technique de l'héritage

permet d'éviter de redéfinir les comportements des nouvelles applications.

3 L'INTERFACE GRAPHIQUE

L'interface permet la saisie du projet ; la saisie doit être aisée et rapide, et ne doit pas être redondante. Les aides à la saisie doivent exister au sein de l'interface : valeurs forfaitaires, liste de valeurs admissibles, tests de cohérence,... Nous présentons ici quelques points qui nous paraissent essentiels pour concevoir et développer une interface graphique dans le domaine de la thermique du bâtiment. La méthodologie d'approche pour concevoir une interface graphique consiste en général à :

- définir tout d'abord les tâches que doivent effectuer les utilisateurs au niveau de l'interface ;
- répertorier les habitudes de travail liées à l'utilisation des logiciels graphiques ;
- répertorier les schémas de raisonnement usuels en thermique du bâtiment afin d'opter pour un mode de dialogue ;
- choisir enfin le type de langage informatique pour le développement de l'interface.

3.1 Rôle de l'interface

L'interface graphique "idéale" doit permettre à l'utilisateur de décrire son bâtiment et d'effectuer un certain nombre d'évaluations ; l'utilisateur doit pouvoir notamment :

- concevoir son projet du point de vue architectural ; il s'agit essentiellement à ce niveau de problèmes liés au respect simultané de la réglementation et des exigences du client ;
- choisir les matériaux de construction et les techniques de mise en œuvre (isolation par l'intérieur, par l'extérieur,...) ;
- choisir les équipements (conditionnement d'air, ventilation, chauffage).

L'utilisateur doit pouvoir aussi :

- introduire les paramètres concernant son projet (valeurs des surfaces, des longueurs, des rendements,...) ;
- introduire les conditions aux limites (températures extérieures, rayonnements,...) ;
- simuler ;
- visualiser les résultats ;
- stocker ses projets.

3.2 Habitudes de travail

Les habitudes de travail liées à l'utilisation des interfaces graphiques sont basées sur :

⁽²⁾ - TRNSYS est un logiciel de calculs thermiques pour le bâtiment ; il a été développé à l'Université de Wisconsin-Madison USA [9].

⁽³⁾ - Le-Lisp est un système LISP, développé à l'INRIA.

⁽⁴⁾ - Aïda est un système de construction d'interfaces graphiques pour le système Le-Lisp ; il a été développé par la société ILOG.

- le multi-fenêtrage ;
- la manipulation directe des objets graphiques.

Les fenêtres sont des objets structurés typés, qui ont des représentations graphiques sur l'écran, sous la forme de cadre rectangulaire muni d'un titre. Les fenêtres peuvent se superposer, une fenêtre peut être partiellement ou complètement cachée par une autre ; elles peuvent être déplacées, redimensionnées, ouvertes ou fermées (représentation sous forme réduite). Les fenêtres sont pilotées par un dispositif de pointage : la souris. Les fenêtres peuvent contenir deux types d'objets graphiques : les images et les "applications". Une image est un objet qui se dessine sur un support, il s'agit d'un élément purement graphique ; une "application" est un objet qui contient des attributs graphiques (police de caractères, couleur de fond, style de ligne, curseur, motif de remplissage), et qui possède un comportement au sein de l'interface. Les objets graphiques communément utilisés sont les boutons poussoirs, les icônes, les éditeurs de texte, les éditeurs d'arbres, les défileurs. Par ailleurs, les fenêtres fournissent un modèle intuitif pour gérer plusieurs activités en parallèle, chaque activité ayant lieu dans une fenêtre.

Le principe de la manipulation directe consiste à représenter les objets de l'interaction et à utiliser des actions physiques sur ces objets (généralement à l'aide de la souris). Les effets de ces actions doivent être visibles et facilement réversibles. Il doit être également possible d'interrompre une tâche pour en effectuer une autre, de mener des tâches en parallèle, ou d'effectuer une sous-tâche ; le multi-fenêtrage autorise ce type de travail. La qualité d'un système à manipulation directe est fortement liée à l'analogie avec le monde réel.

Les "applications" reçoivent des messages lorsque l'utilisateur réalise des opérations avec la souris sur la zone rectangulaire associée. Les actions de l'utilisateur sont appelées événements, dont les principaux sont :

- un bouton de la souris a été enfoncé (clic) ;
- déplacement de la souris en maintenant un bouton enfoncé ;
- relâchement d'un bouton ;
- un caractère au clavier a été frappé ;
- la souris entre ou sort d'une fenêtre.

Il est possible de gérer des événements particuliers, issus de la combinaison des événements cités plus haut ; par exemple, déplacer la souris en maintenant un bouton et un caractère au clavier enfoncés. La gestion de tels événements doit être prévue pour les utilisateurs expérimentés car ils constituent des "raccourcis" utiles pour déclencher rapidement certaines actions.

3.3 Le dialogue Homme/Machine

Dans le domaine qui nous intéresse (thermique du bâtiment), nous retiendrons comme possibilités du dialogue Homme/Machine :

- l'assemblage de schémas-blocs [3] ;
- l'analogie électrique (dialogue fondé sur l'assemblage de

- composants électroniques, résistances, capacités,...) ;
- l'éditeur de texte (dialogue fondé sur l'écriture d'un système d'équations, selon une certaine syntaxe) ;
- le dessin du système à étudier (dialogue des outils de CAO).

Chacun de ces types de dialogue présente avantages et inconvénients, en fonction des utilisateurs, des domaines d'application, et des codes de calculs ; par exemple, pour un mathématicien chevronné, s'exprimer au travers d'équations ne pose guère de difficultés.

Si on opte pour l'assemblage de schémas-blocs [3], l'utilisateur doit :

- déterminer quels composants sont présents ; pour ce faire, l'utilisateur doit, mentalement, décomposer le système à étudier en objets physiques (pompes, tuyaux, murs, zones,...) ;
- déterminer à quels modèles (c'est-à-dire à quels schémas-blocs) ils correspondent ; il faut alors mettre à la disposition de l'utilisateur des bibliothèques de modèles de composants (modélothèques) afin qu'il puisse choisir les modèles qui conviennent à son problème ;
- déterminer quelles relations existent entre les modèles sélectionnés ;
- simuler à partir du système ainsi reconstitué par modules et connexions.

Dans le secteur bâtiment, le dialogue s'effectue en général via le dessin du bâtiment et de ses équipements. Les outils fondés sur ce type de dialogue sont les outils de CAO (Conception Assistée par Ordinateur). Ils sont bien adaptés au mode de travail des différents intervenants du bâtiment ; en effet, à tous les niveaux de l'étude d'un projet de bâtiment est produit un plan graphique (structure, architecture, électricité, ...). Il n'existe pas à l'heure actuelle d'environnement logiciel intégrant à la fois le dialogue des outils de CAO, des modèles thermiques performants en régime variable et des fonctions d'aides "expertes". Ce type d'outil reste à développer.

3.4 Les traducteurs

A l'issue du travail réalisé par l'utilisateur au niveau de l'interface et quel que soit le mode d'interaction, le projet va subir deux transformations transparentes pour l'utilisateur :

- la première transformation, effectuée par un traducteur "générique", consiste à traduire le travail de l'utilisateur en données sous format "neutre" ; la structuration sous un format "neutre" va favoriser la communication entre différents outils de simulation en autorisant la réutilisation des données pour d'autres simulations dans d'autres environnements ; les données sous format "neutre" reproduisent le travail de l'utilisateur, mais ne sont pas exploitables telles quelles par un code de simulation, elles doivent être traitées par des traducteurs "spécialisés" ;
- la seconde transformation, effectuée par un traducteur "spécialisé" (thermique, acoustique,...), va formaliser la

description du problème de l'utilisateur, c'est-à-dire va transformer les données stockées sous format neutre en données exploitables par le code de simulation ; en général, il s'agit de générer un fichier d'entrée écrit dans une certaine syntaxe exploitable par l'outil de simulation. Le contenu de ces fichiers peut soit représenter un assemblage de boîtes noires (fichier DECK pour TRNSYS [9]), soit représenter un réseau analogique (un réseau analogique peut donner lieu à l'expression d'un système d'équations qui peuvent être représentées sous forme matricielle dans le fichier d'entrée [4]).

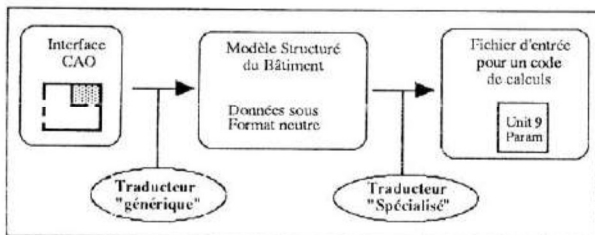


Figure 1 : Exemple de processus de transformation d'objets à l'écran en données exploitables.

3.5 Approche Orientée Objets pour la conception de l'interface

Pour la conception de l'interface, l'approche orientée objets répond aux soucis de réutilisabilité et d'extensibilité. Les programmes sont de nature modulaires grâce à la création d'entités cohérentes et facilement manipulables : les objets.

La question essentielle à laquelle est confrontée un développeur est la suivante "Quels objets sont concernés ?". Cette question touche deux aspects ; d'un côté, elle concerne les objets appartenant à la réalité physique environnante (pompes, régulateurs, bâtiments...) ; d'un autre côté elle concerne les représentations informatiques appropriées, appelées aussi objets. On parlera d'objets externes et internes. Au stade de la conception, il s'agit plutôt de passer en revue les classes d'objets externes dont on veut modéliser le comportement ; au stade de l'implémentation informatique, les systèmes à modéliser doivent être écrits comme des objets internes.

Chaque objet appartient à une classe ; une classe est caractérisée par des méthodes et des champs associés. Une classe est un objet "abstrait" capable d'engendrer d'autres objets appelés instances. Une instance est un objet "concret" ; l'instanciation se fait par duplication de toute la structure de l'objet abstrait : les champs ou méthodes définis au niveau d'une classe se retrouvent dans tous les objets "concrets" instanciés à partir de cette classe. Les objets appartenant à une même classe ont la même forme, ont un comportement commun, mais sont différents par la valeur de leurs champs. Appliquer une opération à une instance consiste à lui envoyer un message contenant le nom de la méthode devant être activée.

Associée à ce concept de classe, se rattache une notion importante : l'héritage. Il est parfois judicieux de regrouper au niveau d'une classe "mère", appelée superclasse, un cer-

tain nombre de propriétés communes à plusieurs classes. Toutes les classes, sous-classes de cette classe mère, hériteront de ces propriétés communes (champs et méthodes). L'ensemble des classes forment ce qu'on appelle un graphe d'héritage. En général, l'héritage des variables est statique : les variables héritées sont recopiées dans la structure représentant la classe. Par contre, l'héritage des méthodes est dynamique : les méthodes héritées ne sont pas recopiées et la méthode à appliquer lors d'un envoi de message est recherchée dynamiquement dans le graphe d'héritage.

La conception d'une interface graphique est facilitée par l'utilisation des langages orientés objets ; en effet, une interface graphique met à profit les caractéristiques des langages orientés objets :

- la structuration en objets se justifie puisqu'on a à représenter des entités (planchers, bardages, toitures, cloisons,...) ayant des données propres et des comportements propres ;
- l'instanciation est particulièrement utile puisqu'on a de nombreux objets du même type à mettre en place pour l'élaboration d'un projet de bâtiment ;
- l'héritage trouve son utilité puisque le développeur d'une interface graphique a souvent à construire de nouveaux types d'objets à partir de ceux existants ; pour l'utilisateur, l'héritage trouve son utilité dans la construction des bibliothèques de modèles, appelées aussi modélothèques.

3.6 Les bibliothèques

L'archivage des données et des résultats doit être possible au sein de l'interface. Pour cela, nous distinguons deux types de bibliothèques : les bibliothèques de modèles et les bibliothèques de projets.

Si le modèle thermique du système Bâtiment/Equipements est généré de façon automatique par les traducteurs "spécialisés" (à partir du dessin du bâtiment par exemple), alors pour le thermicien seules des bibliothèques de projets sont indispensables. L'utilisateur de l'interface a cependant besoin, pour la conception de son projet, de puiser dans une bibliothèque un certain nombre d'objets : murs, poteaux, poutres,... ; ces objets qui peuvent être graphiques ne représentent nullement des modèles de comportement thermique de composants ; on ne parlera donc pas de modélothèques pour désigner les bibliothèques d'où sont issus des objets purement graphiques, ou possédant une sémantique pauvre.

Par contre, si le passage Bâtiment → Modèle thermique du système Bâtiment/Equipements n'est pas automatisé, il faut que le thermicien puisse décrire le comportement thermique du système Bâtiment/Equipements par des schémas-blocs. La présence de modélothèques au niveau de l'interface est alors indispensable.

Il est pratique de présenter les bibliothèques sous la forme d'arbres (Figure 2), dont les nœuds sont, soit des dossiers, soit des modèles ou des projets. Les dossiers apparaissent sous la forme d'un texte ; les modèles ou les projets ont la forme d'icônes graphiques.

Cette représentation présente plusieurs avantages :

- ◆ elle simplifie l'accès aux objets, par la visualisation de la totalité de la bibliothèque ; le nombre d'opérations nécessaires à un utilisateur avec le dispositif de pointage pour accéder à un objet est faible ;
- ◆ elle permet d'obtenir des bibliothèques structurées (à défaut ordonnées) et personnalisées assez rapidement .

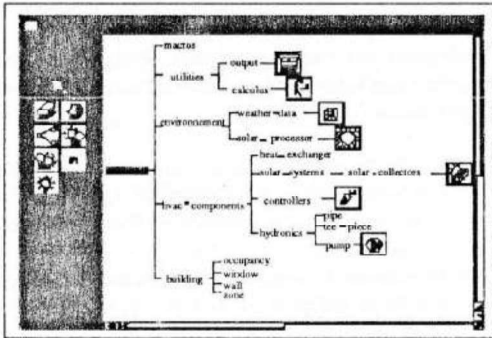


Figure 2 : Exemple de bibliothèque de modèles.

Deux approches permettent d'obtenir des bibliothèques structurées :

- une approche non orientée objet (où la notion d'héritage n'existe pas) ; dans cette approche, le scénario le plus fréquent pour la construction rapide d'une bibliothèque est le suivant. Le développeur de modèles (DM) a pour objectif par exemple l'insertion de deux modèles M2.0 et M2.1, qui ont la particularité de se ressembler, au sein d'une bibliothèque. Pour y arriver, il doit d'abord créer le modèle M2.0, il crée ensuite un modèle M2.1 duplicata du modèle M2.0, puis adapte le modèle M2.1 ; afin de mieux structurer sa bibliothèque, le DM peut créer un nœud M2, sous lequel sont rattachés les modèles M2.0 et M2.1. L'inconvénient dans cette approche réside dans la modification d'objets existants : si l'utilisateur veut modifier une propriété commune à tous les objets rattachés au nœud M2, il devra réaliser autant d'opérations "Modification d'objet" qu'il y a d'objets rattachés au nœud M2.
- une approche orientée objet ; dans cette approche, les nœuds de la bibliothèque sont des classes ; l'utilisateur se pose la question suivante : quelles sont les propriétés (attributs et méthodes) communes à tous les modèles d'un même groupe que je peux définir au niveau d'une classe ? La réponse à cette question doit conduire le DM à concevoir la classe M2. Ensuite, le DM crée les modèles M2.0 et M2.1 et bénéficie de tout ce qui a été préalablement défini pour la classe M2, à partir du moment où le DM rattache M2.0 et M2.1 à M2. Cette approche demande un effort de synthèse de la part du développeur de modèles, si celui-ci veut utiliser de façon optimale les propriétés d'héritage.

En résumé :

- pouvoir bénéficier des propriétés d'héritage dans les

modélothèques est particulièrement utile ; l'héritage est un avantage pour le développeur de modèles puisque beaucoup de modèles ont de nombreux points en commun ; cela va notamment permettre d'assurer l'extensibilité du logiciel, en autorisant des modifications rapides et sûres des objets stockés ;

- pouvoir bénéficier des propriétés d'héritage est moins justifiée dans les bibliothèques de projets ; en effet, à l'heure actuelle, il n'existe pas de structuration reconnue associée aux projets de bâtiment.

4 FONCTIONS D'AIDES "EXPERTES"

Pour améliorer la productivité de l'utilisateur, les fonctions d'aides expertes doivent être conçues autour de trois idées :

- ✓ mettre en place des bases de données ;
- ✓ mettre en place des guides spécialisés ;
- ✓ intégrer le raisonnement par cas dans l'environnement de simulation.

4.1 Les bases de données

Les bases de données vont aider l'utilisateur dans la conception de son projet. Les bases de données d'un ESI pourraient être les suivantes :

- ◆ une base de données qui contient les données climatiques de référence (issues de la réglementation) ;
- ◆ une base de données qui contient les données qui concernent les matériaux (caractéristiques techniques moyennes, coûts, lieux de production, lieux de distribution,...) ;
- ◆ une base de données qui contient les données (puissances, rendements, coûts, lieux de distribution,...) qui concernent les équipements de chauffage et de climatisation ;
- ◆ une base de données qui contient les données (caractéristiques techniques moyennes, coûts de mise en œuvre, lieux de production,...) relatives aux ouvrages tels que les fenêtres, les panneaux préfabriqués (préfabrication lourde), les façades légères, les éléments de remplissage.

• Utilisation possible d'une base de données pour un problème d'optimisation

Considérons le problème suivant ; l'utilisateur veut optimiser l'isolation thermique d'un bâtiment ; supposons que l'on ait neuf postes d'isolation (voir figure 3).

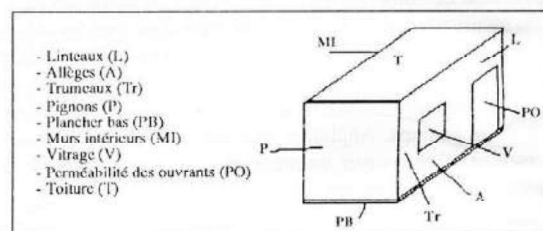


Figure 3 : Postes d'isolation considérés.

Les postes d'isolation sont considérés indépendamment les uns des autres. Pour chaque poste, plusieurs valeurs de la conductance K peuvent être envisagées en faisant varier l'épaisseur ou la nature de l'isolant, la nature du vitrage (simple, double,...) et le type de menuiseries. En vue d'effectuer une optimisation technico-économique, il est possible d'associer à chaque poste une fonction de calcul de prix unitaire. Le but du problème est de déterminer une solution optimale, c'est-à-dire de définir les degrés d'isolation pour chacun des postes. Si la solution est acceptable du point de vue réglementaire, la solution est dite admissible (elle n'est pas forcément optimale).

Soit i le numéro du poste, j le degré d'isolation du poste, C_{ij} le coût unitaire associé (prix unitaire par m^2), Si la surface du i ème poste, le coût d'une solution est donné par la formule suivante :

$$\text{Coût} = \sum_{i=1}^9 S_i \times C_{ij}$$

Les méthodes d'optimisation utilisées sont en général des méthodes de recherche opérationnelle ; ce problème d'optimisation peut être entièrement automatisé dans un environnement de simulation à condition de trouver toutes les informations nécessaires dans des bases de données.

4.2 Les guides spécialisés

Les guides spécialisés ont pour objectif d'aider l'utilisateur de l'environnement de simulation à concevoir son projet ; ces guides doivent être conçus pour prendre en charge tout ou partie des mécanismes de raisonnement qui sont utilisés habituellement dans le cadre d'un outil de simulation (en thermique du bâtiment dans notre propos). Ces guides peuvent être formalisés sous forme de systèmes experts (appelés aussi systèmes à bases de connaissances).

• Le concept de système expert

Le concept de système expert regroupe deux notions qui sont :

- les connaissances nécessaires pour résoudre un problème ;
- les mécanismes de raisonnement exploitant ces connaissances, appelés aussi moteurs d'inférences. Le noyau d'un système expert comprend trois parties :
 - ✓ une base de règles qui constitue la connaissance permanente ;
 - ✓ un moteur d'inférence qui est le mécanisme de raisonnement exploitant ces règles ;
 - ✓ une base de faits qui représente la mémoire de travail du système.

Une règle est généralement rédigée de la façon suivante :

[Si Conditions alors Conclusions (B)]

La partie "conditions" (ou prémisse) doit être vérifiée

pour que la règle s'applique. La partie "conclusions" correspond à l'action déclenchée lors de l'activation de la règle. Le coefficient B traduit l'incertitude sur la connaissance exprimée par la règle, appelé degré de vérité ou coefficient de vraisemblance.

Le problème essentiel des systèmes experts concerne la maintenance de la base de connaissances (base de règles), qui passe par l'insertion de nouvelles règles. Cette remise à jour de la base de connaissances peut engendrer des incohérences telles que le risque de répéter une règle existante, le risque de conflits, l'écriture de prémisses de règles insuffisamment réduites.

• Guides spécialisés d'un ESI

Pour aider au travail de conception, on peut imaginer le développement des fonctionnalités suivantes :

- un guide de conception bioclimatique ; la conception bioclimatique prend en compte la construction et son environnement immédiat ; les règles produites dans ce guide feraient principalement référence à la zone climatique du projet, au relief immédiat, aux exemples de solutions conçues par des centres spécialisés ;
- un guide pour le choix des matériaux ; les règles produites dans ce guide feraient principalement référence aux contraintes architecturales (aspect), à la réglementation en vigueur, à la proximité des lieux de production des matériaux et à la capacité financière du client ;
- un guide pour le choix des systèmes techniques (chauffage et climatisation) ; les règles produites dans ce guide feraient principalement référence à l'aménagement du volume intérieur, à la réglementation en vigueur, à la proximité des lieux de fabrication des équipements et à la capacité financière du client ;
- un guide pour le choix des techniques de mise en œuvre ; les règles produites dans ce guide feraient principalement référence à la proximité des entreprises de réalisation, à la réglementation en vigueur et à la capacité financière du client.

Il est bien sûr possible de mettre en place un seul guide qui regrouperait les diverses connaissances sus-citées.

4.3 Le raisonnement par cas

Cette démarche est courante dans les bureaux d'études. En effet, un ingénieur en bureau d'études a souvent tendance à chercher les informations qui lui manquent dans un projet antérieur, ou à concevoir un nouveau projet à partir de projets déjà existants en les modifiant, lorsque cela est possible : le raisonnement par cas s'inscrit dans cette logique.

Le raisonnement par cas est fondé sur la mise en œuvre de mécanismes de correspondances entre, d'une part, des "objets-solutions" qui représentent des solutions testées antérieurement, et, d'autre part, l'objet "demande" de l'utilisateur qui représente le problème posé. L'objectif est de retrouver parmi les "objets-solutions" le (ou les) cas "les plus similaire(s)" au problème posé. Pour pouvoir réaliser le raisonnement par cas, il faut posséder dans une base de

connaissances un certain nombre de "cas types" ; chaque cas doit être représentatif d'une catégorie de problèmes dont l'utilisation simplifie le raisonnement. Dans le cadre d'un environnement de simulation, les bibliothèques de projets constituent tout naturellement cette base de connaissance (un projet représente en principe une solution testée).

Le système intelligent basé sur le raisonnement par cas s'appuierait alors autour des étapes suivantes :

- ▼ l'utilisateur doit d'abord décrire son problème en dialoguant avec la machine ; le problème à traiter est transformé en un objet structuré possédant des attributs ; ces attributs sont ensuite "triés" pour définir des caractéristiques spécifiques ;
- ▼ les caractéristiques déterminées à partir de l'étape précédente sont utilisées pour rechercher dans les bibliothèques de projets un projet similaire ; cette recherche nécessite la définition de critères de comparaison ;
- ▼ une fois le projet "similaire" déterminé, il est modifié de façon à tenir compte des spécifications du problème à traiter.

Les langages orientés objets sont particulièrement bien adaptés pour automatiser le raisonnement par cas.

L'efficacité de ce type de raisonnement dépend des cas sur lesquels il s'applique. Il s'agit de répertorier les cas de façon "intelligente" : ils doivent être suffisamment généraux pour permettre leur utilisation dans diverses situations, mais ils doivent être suffisamment spécifiques pour éviter les conflits dans la recherche d'un cas similaire et pour couvrir le domaine d'application.

L'intérêt du raisonnement par cas se situe à plusieurs niveaux :

- ▼ le raisonnement par cas permet de compléter les méthodes de raisonnement classiques, notamment dans les domaines où la formalisation est délicate ; en effet, un cas synthétise un savoir-faire touchant à plusieurs domaines ; par conséquent, un cas peut s'avérer être l'équivalent de dizaines de règles de production ;
- ▼ les outils basés sur l'exploitation des cas permettent de constituer et de faire évoluer une mémoire collective d'une communauté d'individus : les utilisateurs des codes de simulation lèguent en général aux futurs utilisateurs essentiellement des projets réalisés, plutôt que des règles de conception bien formulées.

5 CONCLUSION

A l'heure actuelle, dans le domaine de la thermique du bâtiment, il est difficile de parler d'un environnement de simulation qui réponde aux préoccupations de tous les utilisateurs potentiels, compte tenu de la diversité des solutions. En fait, trois questions restent posées :

- ✓ Quels types de modèles mathématiques faut-il insérer ?
- ✓ Quel type de dialogue Homme/Machine faut-il mettre en place ?
- ✓ Que type d'aide faut-il mettre à la disposition des utilisateurs ?

Des éléments de réponse ont été apportés dans les travaux de recherches qui ont porté sur la conception d'un ESI [7], [8], [3]. Ces travaux ont essentiellement porté sur la mise au point d'interfaces "intelligentes" (projet IISIBât). L'application IISIBât, fondée sur la modélisation par assemblage, pose des difficultés pour les utilisateurs "non spécialistes", notamment lors de la recherche des modèles les mieux adaptés au problème posé, et lors de leur connexion. Pour éviter ces tâches (recherche de modèles, connexion des modèles), il faut pouvoir générer de façon automatique le modèle thermique du bâtiment. Une solution consisterait à réaliser un traducteur "spécialisé" qui effectuerait les tâches suivantes :

- ▼ discrétisation du bâtiment en nœuds ;
- ▼ affectation des flux de chaleur aux nœuds pour prendre en compte les apports internes, et les apports gratuits ;
- ▼ mise en forme des équations, et formalisation de ces équations dans un fichier.

Nous appellerons l'environnement conçu à partir de cette approche ESTBât : Environnement pour la Simulation Thermique du Bâtiment. Une application concrète de ce nouveau concept pourrait être la suivante :

- utiliser le formalisme de l'analogie électrique pour poser les problèmes en thermique du bâtiment ;
- introduire un dialogue Homme/Machine fortement inspiré des outils de CAO pour décrire le projet, ce qui va faciliter à moyen terme l'appropriation des outils de simulation par les bureaux d'études ;
- mettre en place des fonctions d'aides expertes construites autour de bases de données, de guides spécialisés, du raisonnement par analogie.

La figure 4 schématise les différents éléments qui pourraient entrer dans un environnement de simulation de type ESTBât.

BIBLIOGRAPHIE

- [1] P. Bacot : "Méthode Modale". Article ; in *Energétique des Bâtiments*, Tome 2 ; PYC édition ; 1988.
- [2] K. El Hassar : "Module Ensoleillement". MODSOL ; Rapport, Sophia Antipolis, CSTB, MGL/90-1164/NB, 1990.
- [3] K. El Hassar : "Conception d'un environnement de simulation intelligent". Application à la thermique du bâtiment ; Thèse, Ecole Nationale des Ponts et Chaussées, Paris, 1992.

- [4] L. Laret : "Modélisation du comportement thermique de l'immeuble expérimental de la DETN". Rapport Final ; Etude, Sophia Antipolis, CSTB MGL 1075, 1989.
- [5] L. Laret : "Le concept de modèle adapté. Séminaire spécialisé "Outil d'aide à la conception et à la gestion"". Article, Sophia Antipolis, CSTB, 1989.
- [6] J. Lebrun : "Modélisation et Désign des systèmes de chauffage, ventilation et conditionnement d'air". In Climat 2 000, Tome 1, Liège, 1985.
- [7] R. Pelletret : "IISIBât : un environnement de simulation intelligent". S. Soubra ; Document de travail, Sop Antipolis, CSTB, 1992.
- [8] S. Soubra : "Concepts et outils pour le développement d'un Environnement de Simulation Intelligent". Thèse Ecole Nationale des Ponts et Chaussées, Paris, 1992.
- [9] TRNSYS : "A Transient System Simulation Program Manual, Solar Energy Laboratory, Univ. Of Wisconsin USA 1990.

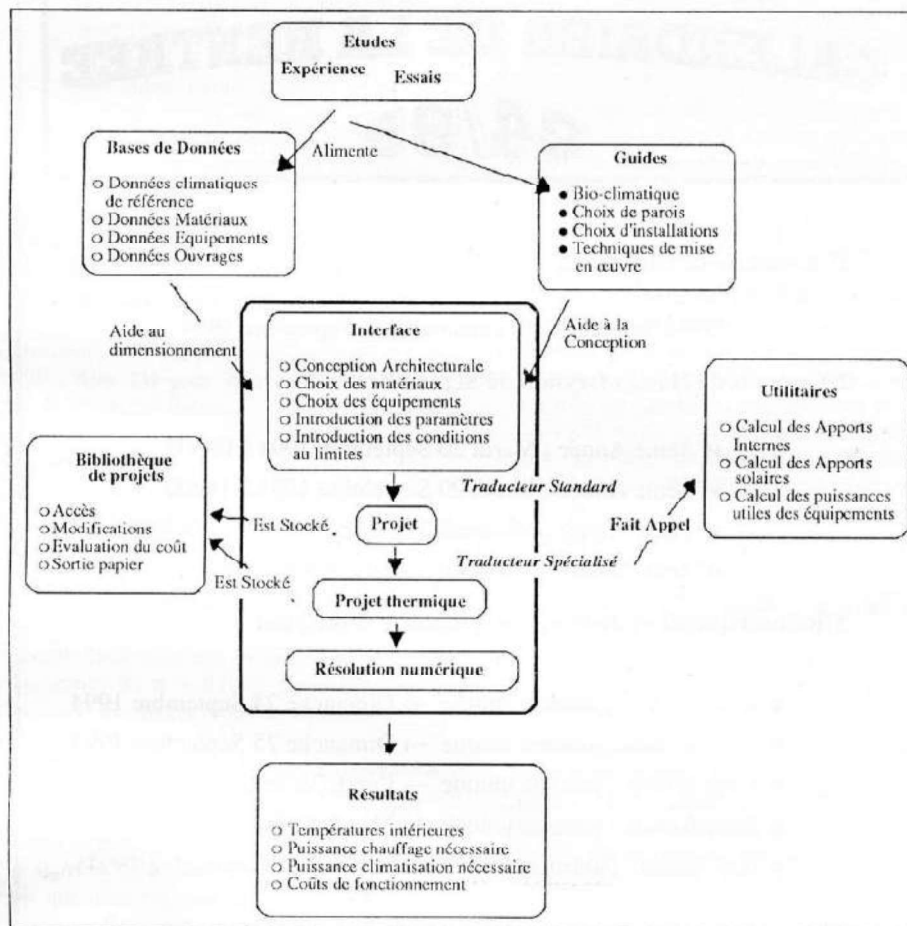


Figure 4 : Projet ESTBât.