A modular approach for developping a modeling environment of thermal systems: the SYMBOL project

G.Lefebvre, R.Ebert, B.Flament
Groupe Informatique et Systèmes énergétiques *

Abstract: we present the SYMBOL project which is based on a modular approach of modeling. The main objectives of this project is to develop a coherent set of modeling tools which can be assembled in order to simply build particular and adapted modeling programs and to allow handling of "natural" concepts during the modeling process. The paper will deal mainly with the modularity of the SYMBOL'environment, and the needed formalism and abstraction effort made in the SYMBOL project.

1 Introduction

As occured in a lot of laboratories involved in thermal systems modeling, the simulation and analysis tools we have been developing until a recent time were strongly adapted for studying particular thermal systems and very specific of the employed modeling principles. These tools, designed and maintained by distinct persons in different — and often incompatible — computing environments, were similar for about 60% of their theoretical contents. Each new thermal system or new modeling process obliged to design a new data structure, to build new algorithms, and to write entirely a new code.

The raise of new logical concepts and programming tools induced a new modeling approach, in particular when it is applied on objects in which the structure and the pertinence of the information are as important as the numerical quality of the results. In that case, a model must be able to predict accuratly the evolution of the physical state of the real system, but also to provide a data structure characterizing and explaining — from the physical point of vue — the way it responds to excitations. A building, *thought* as a physical system and particularly a thermal one, is a good example of *systemic object.*

We will first introduce the modeling principles upon which are designed all the SYMBOL elementary tools. Then, we will describe two particular modeling processes which can be followed in the SYMBOL environment: *modal transformation* and *object oriented simulation.*

*GISE - ENPC - EMP La Courtine 93167 Noisy le Grand FRANCE – email: gl@enpc.fr

2 Modeling principles of the SYMBOL project

2.1 Aims

Two aims have oriented the theoretical development of the SYMBOL project. 1 – The SYMBOL environment should be developed as a coherent set of compatible elementary bricks which could be combined in order to fashion particular and suited modeling tools. It is obvious that a systemic point of view should be choosen in order to reduce the apparent complexity of physical objects by structuring and rationalising their representation. This first choice might convert the modeling process in an easier and more flexible one. Models of systems or sub-systems must be reusable in order to capitalize the knowledge in libraries and to save repetitive modeling efforts. 2 – The SYMBOL environment should provide tools for handling *natural* concepts which belong to the *reality* of the engineer (and avoiding the ones of the physicist, the computer scientist, the mathematician or the numerician).

2.2 Means

The means available to reach our objectives are based mainly on *modularity* and *visibility* rules of the computing codes and of the modeling structure.

Splitting a code in *modules* allows to reuse existing ones to build new programs. The modules, which can be written by different people and shared among various projects, are better designed, documented, validated because the initial effort is redeemed by each reuse. Visibility rules allow to separate which part of a module may be seen by other ones — what it does — from the part which is hidden from the rest

of the environment — how it does it —. This code modularity is the concrete support of a more conceptual one, the modeling modularity. A first modeling modularity consists in splitting a system representation in coupled sub-systems following a recursive process until the *terminal* sub-systems can be represented by simple elementary — and by the way not "splittable" — models. Another one supports a specialisation mecanism allowing the implementation of an abstraction – concretisation concept. We will now precise and illustrate these ideas with the classical exemple of heat conduction in a multilayer wall. We will try to express the representation of this wall in the various hierarchical world as defined in [4].

2.3 Example

We want to describe the thermal behaviour of a wall — made of two homogeneous layers and excitated by convection exchanges — in order to predict the evolution of its thermal state. Figure 1 can be seen as an image of the "real" wall.

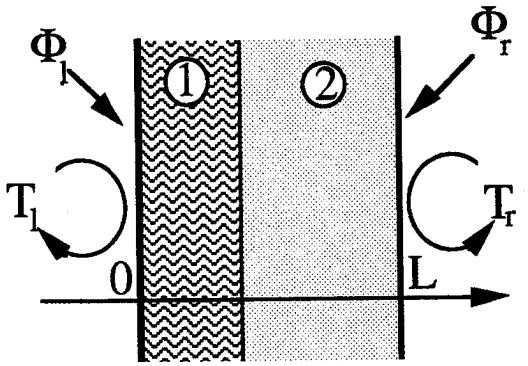

Figure 1: "Usual" representation of a "real" multilayer wall

The technical world The multilayer wall is "obviously" made of four *subsystems* (C_1, C_2, C_3 and C_4) coupled by three *interfaces* (I_1, I_2 and I_3) (Figure 2).

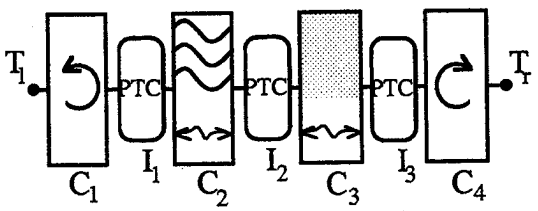

Figure 2: Multilayer wall seen in the "technical" world

The three interfaces describe the same kind of coupling, a perfect thermal contact (PTC). The subsystems, as they appear in Figure 2, are *terminal* models. C_1 and C_4 describe convective phenomenas on both sides of the wall. C_2 and C_3 describe heat conduction in a layer. Figure 1 is an unstructured and poor representation. The highest pertinent abstract level at which the engineer will refer corresponds to the one illustrated in Figure 2. At this stage, physical phenomenas, component limits, coupling structure and nature can be specified (but not defined, this belongs to the physical world).

The coupling graph is arbitrary. As a matter of fact, it is possible to design various coupling graphs, as shown in Figure 3.

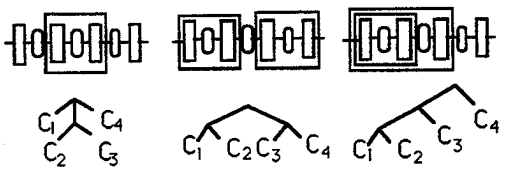

Figure 3: Different possible coupling graphs in the technical world

The physical world At a more concrete level, the physical one, are specified the physical laws and hypothesis inside the components and interfaces.

The frontiers are the visible parts of the models and are coupled as a whole in an interface. For example, the perfect contact interfaces are ones in which must be verified the equality of the neighbouring temperatures and the compatibility of the respective fluxes (algebraic sum equal to zero) (Figure 4).

The models describing C_1 and C_4 are *instanciations* of *generic* convective models which could be as follows:

$$\Phi_{i-j} = \alpha S (T_i - T_j) \qquad (1)$$

where Φ is the thermal flux passing from side i to side j, and T_i (resp. T_j) is the temperature of side i (resp.j), i and j being one the air and the other the wall face. The instanciation consists in giving particular values to the model parameter, α and S in (1). Following a same reasoning, C_2 and C_3 are instanciations of a conduction model, which could consist in the Fourier conduction law and Diriclet boundary conditions:

$$\begin{aligned} \vec{\varphi}(M,t) &= \lambda \vec{\nabla} T \\ T(0,t) &= T_l(t) \\ T(L,t) &= T_r(t) \end{aligned} \qquad (2)$$

It must be noticed that physical laws are expressed at this stage as implicit equations; they are only equality relations which must be verified by the set of variables and parameters of the models, but the calculation methods to employ in order to verify these relations are not yet defined.

The mathematical world In the mathematical world can be made some mathematical transformations, for example a modal one [10], in which a layer

– conduction problem can be expressed as:

$$\begin{aligned} \frac{d\vec{X}}{dt} &= F\vec{X} + B\frac{d\vec{U}}{dt} \\ \vec{T} &= P\vec{X} + G\vec{U} \\ \vec{Y} &= H\vec{X} + S\vec{U} \end{aligned} \quad (3)$$

in which $\vec{U}$ contains the excitations variable (for example the temperature of both surfaces of the layer), $\vec{T}$ contains the internal temperatures, and $\vec{Y}$ contains the outputs of the model, for exemple the fluxes going out the layer through each face. F, B, H, P, G and S are parameters of the model which must be instanciated for each particular modeled layer. Their particular signification will be explained later.

The *pins* are unknown variables (inputs or outputs) of the models which can be *connected* to the ones of other models. In the physical world, pins were gathered in *frontiers*. For example, in Figure 4, $F_2^{C_1}$ is a frontier of component C_1 which gathers the two pins $T_2^{C_1}$ (temperature of face 2) and $\Phi_2^{C_1}$ (flux going out through face 2).

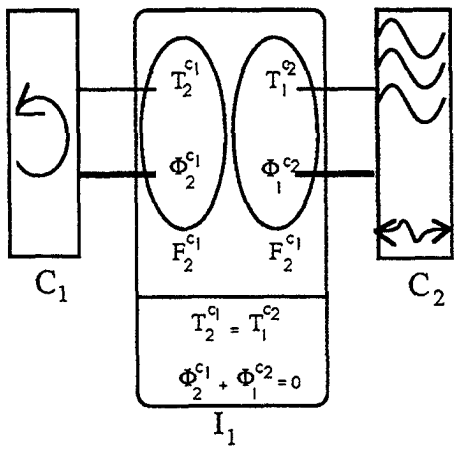

Figure 4: Coupling frontiers in an interface

Numerical world Numerical methods which remain necessary to get a solution (as discretisation or sampling) are applied at this stage. As the physical laws choosen at the physical stage depend on the simplification hypothesis, the numerical methods depend on the objectives of the modelling process and the required accuracy and detail level.

We will now focus on two particular uses of this modeling scheme: the modal transformation and the object oriented deterministic simulation.

3 Architecture of the SYMBOL environment

The SYMBOL environment is a set of autonomous programs communicating by files whose contents respects a particular syntax based on the modeling concepts described here above. It can be seen in Figure 5

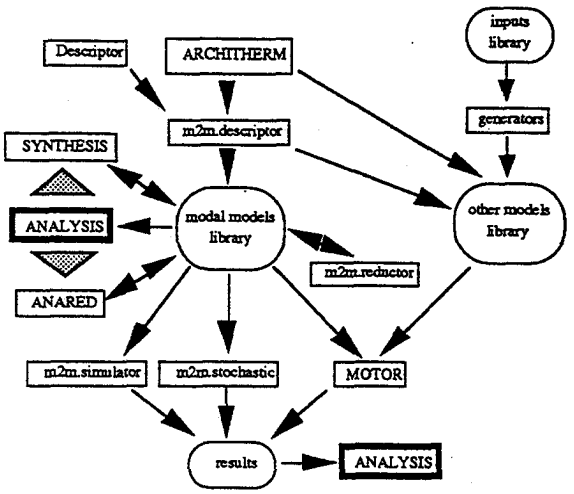

Figure 5: Architecture of the SYMBOL environment

that there is a main orientation from the description of a problem towards results of simulation or information obtained from analysis.

A lot of modeling processes are possible in the SYMBOL environment, but a common one to model a building would be the following. A graphical-aided description of a building would be made with ARCHITHERM which outputs are files containing all detailed thermophysical and geometrical properties. They are used by m2m.descriptor to build a multizone modal model of this building. The modal model can then reduced with ANARED or m2m.reductor, using the information provided by the various available ANALYSIS tools. The reduced modal model can then be used to perform a deterministic modal simulation with m2m.simulation, or a stochastic one with m2m.stochastic, or to be coupled with non - modal models in an object oriented simulation with MOTOR.

m2m.descriptor can also produce one modal model per zone plus the model describing the coupling between them. These models can be used by SYNTHESE in order to build a unique modal model of the whole building, or by MOTOR to simulate non - linear heat exchanges.

4 Modal transformation in the SYMBOL environment

4.1 The modal approach in thermal modeling

The modal method is very powerful to obtain an accurate representation of the thermal dynamic behaviour of linear systems. These methods, well-known in mechanical engineering (vibration modes computing), have been developed for 10 years now in thermal engineering in order to get easily accurate reduced models. The method can be applied to systems which involve conductive exchanges, enthalpy transport, and radiative heat transfers. Details about

the modal method can be found in [11, 5].

In practice, building a modal representation of a thermal system needs first to split the temperature field into a pseudo-static term T^g and a dynamic one T^d:

$$T(M,t) = T^g(M,t) + T^d(M,t) \qquad (4)$$

The dynamic term is then decomposed on a basis made of the eigenfunctions $V_i(M)$ of the spatial heat operator; in the same way the pseudo-static term is expressed as a function of elementary static solutions $S_p(M)$. Thus, the above expression becomes:

$$T(M,t) = \sum_p S_p(M)U_p(t) + \sum_{i=1}^{N} V_i(M)X_i(t) \quad (5)$$

where $U_p(t)$ represents the time varying driving force range and N the number of eigenelements kept.

The excitation state X_i of the ith-order eigenmode verifies the first-order differential equation :

$$\frac{dX_i(t)}{dt} = z_i\, X_i(t) + \sum_p b_{ip}\dot{U}_p(t) \qquad (6)$$

where z_i is the ith-order eigenvalue associated with the ith-order eigenfunction $V_i(M)$ and b_{ip} is the result of a scalar product between the p-order elementary static solution $S_p(M)$ and the ith-order eigenmode.

To select the N eigenmodes that we want to keep, we have developed some tools (ANARED software) which gather some classical reduction methods [12, 13] and some new ones (modal amalgam [16]).

A N-order modal model is then represented by the set of equations (5) and (6). Its matrix form is given by the relation (3).

One can see that the modal method allows to get easily a reduced model if we know the eigenelements (eigenmodes plus eigenvalues) and particularly the N eigenmodes that we want to keep. In general to compute these eigenelements we have to solve a matrix eigenvalue problem. The dimension of this problem corresponds to the number of discretization nodes. This numerical step is a limitation of the modal method when we want to study complex systems (multizone buildings, 3D thermal bridges...), because the discretization involves a lot of nodes. As a matter of fact, to keep only a few dominant eigenmodes, we have to diagonalize big matrices (1000, 10000...), which needs computers with high computing capacity and a lot of memory. Thus, to avoid this problem when studying complex systems by the modal approach, we have developed methods based on the modular structure of the SYMBOL environment: a complex thermal system is described as the assembling of various components being coupled. The eigenfunctions of the spatial heat operator defined in the whole system (called *global eigenfunctions*) are computed from the eigenfunctions of the same operator but defined in its components assumed to be independant (called *local eigenfunctions*). This procedure described briefly above corresponds to a dynamical substructuring approach also called *modal synthesis method.*

In the following, we present the modal synthesis method with the help of two examples, based on the modularity concepts introduced in the first part of this paper. More details can be found in [1, 6, 7, 8].

4.2 A particular application of the SYMBOL modeling structure : the modal synthesis

Case of a multilayer wall Let us consider again the two - layers wall described figure (1), and let us compute the eigenfunctions $\mathcal{V}_{ki}$ and the eigenvalues z_i of this thermal system by the modal synthesis.

The latter consists in solving the global eigenvalue problem, the solutions of which are $\mathcal{V}_{ki}$ and z_i, using the modular description presented in the § 2.

In the *technical world*, the two - layers wall can be represented by two components coupled by one interface which describes a perfect contact. Let us notice that for this simple exemple, only one coupling graph is possible since we do not consider the external heat exchange.

In the *physical world*, the components are represented by generic models of conduction and the interface model can be illustrated by Figure 4.

Until now, no specificity related to modal synthesis has been necessary. Specific concepts of this method are introduced in the mathematical and numerical worlds.

In the *mathematical world*, we give the mathematical translation of the substructuring problem to solve. In this world, for the modal synthesis, each component is represented by equations related to the global eigenvalue problem:

$$\begin{aligned} \nabla K(M)\nabla(\mathcal{V}_{ki}(M)) &= z_i\, C\, \mathcal{V}_{ki}(M) \\ B(\mathcal{V}_{ki}(M)) &= 0 \end{aligned} \qquad (7)$$

where k is the component index, B a boundary conditions operator [5], K a thermal conductivity and C a volumic capacity. In the same way, the interface model is represented by two equations defined on the linked frontiers, as follows :

$$\begin{aligned} \mathcal{V}_{1i} - \mathcal{V}_{2i} &= 0 \\ -K_1\,\nabla(\mathcal{V}_{1i}).\vec{n}_1 - K_2\,\nabla(\mathcal{V}_{2i}).\vec{n}_2 &= 0 \end{aligned} \qquad (8)$$

We can notice that for this "1D"[1] system, these two relations are scalar equations.

In the *numerical world*, we precise the solving methodology of the problem given in the mathematical world. To solve the latter by modal synthesis, it is necessary to introduce a new concept of *coupling*

[1] ... in the modal synthesis context. That is to say that a system can be considered 1D for the modal synthesis, if the different linked-up frontiers can be represented by one temperature and one heat flow density.

functions [8]. As a matter of fact, we can show (Figure 6) that the global eigenfunctions can not be reconstitute only from local eigenfunctions which may vanish on the linked frontiers[2]. Thus the modal synthesis consists in writing for each k component, the global eigenfunctions as follows :

$$\mathcal{V}_{ki}(M) = \sum_l p_{kli}\, V_{kl}(M) + \beta_{ki}\, \tilde{S}_k(M) \quad (9)$$

V_{kl} is the l-order eigenfunction of the heat operator defined in the k component and $\tilde{S}_k$ is the coupling function of the k component. For "1D" systems, the latter are defined as elementary static field related on driving forces applied on the linked up frontiers.

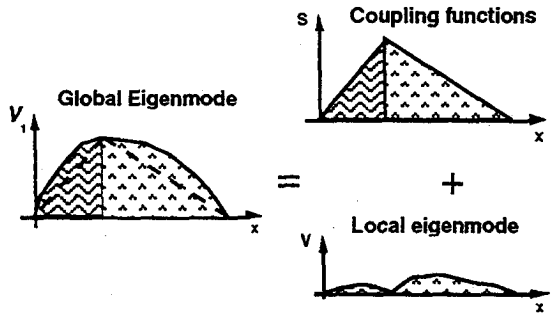

Figure 6: Global eigenfunctions reconstitution for "1D" systems

In practice, the serie is necessarily truncated and the above relation is an approximation of the global eigenfunctions[3]. The relation (9) provides a representation of the components and of the interface in the numerical world deduced from their representation in the mathematical world. As a matter of fact, we can show that from the relations (7) and using the orthogonality property of the local eigenfunctions, we obtain a new equation $f(p_{ki}, \beta_{ki}) = z_i\, g(p_{kli}, \beta_{ki})$ where f and g are functions depending on the decomposition coefficients. This new relation corresponds to the representation of the k component in the numerical world. In the same way, from the expressions (8), we get an explicit relation between the decomposition coefficients p_{1li}, p_{2li}, β_{1i} and β_{2i} which corresponds to the representation of the interface in the numerical world.

The physical coupling between components is then translated in the numerical world by solving the two last equations.

Case of multidimensional thermal systems

The main difference between "1D" systems and multidimensional ones lies in the description of the coupling interface in the mathematical and numerical worlds. In the following we give some indications about the coupling representation in the case of a perfect thermal contact.

[2]Let us recall that the eigenfunctions verify homogeneous boundary conditions.

[3]The interest of the modal synthesis is to get a good approximation of $\mathcal{V}_{ki}$ with only a few local functions.

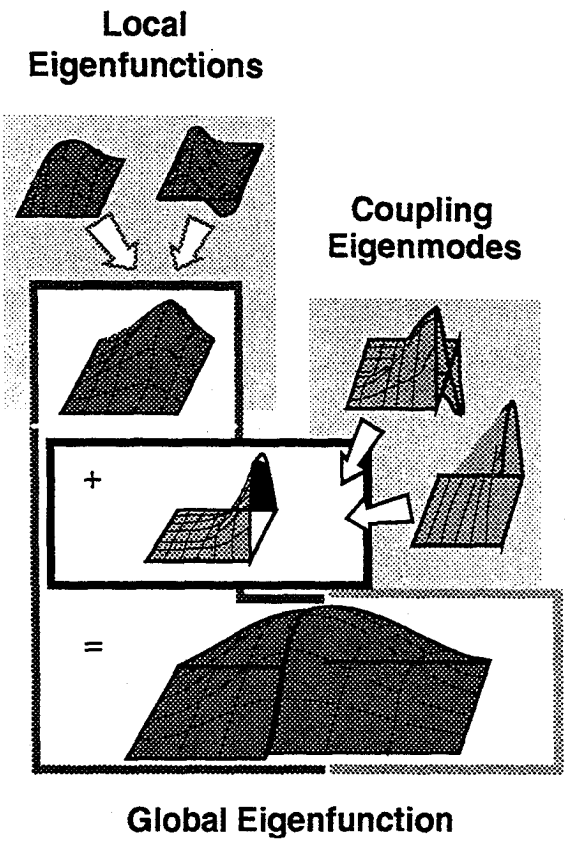

Figure 7: Global eigenfunction reconstitution for multidimensional systems

In the mathematical world, the relations describing the coupling between components are not scalar ones any more, but are functional equations verified for each point of the linked frontiers.

In the numerical world, in order to solve the problem described in the mathematical world, it is also necessary to introduce *coupling functions* [8] for the same reasons as those given in the last paragraph. However these functions are now defined as the solution of an eigenvalue problem related on a link-up operator close to a Poincare Steklov operator (well-known in domain decomposition to solve static problems [3]) and defined for each component. They are called *local coupling modes* (Figure 7). Thus the modal synthesis consists in writing an approximation of global eigenfunctions as equation (9) where a serie of coupling eigenmodes is put in place of the coupling function.

Due to the fact that, in the numerical world, the series are truncated and the continuous functions discretizated, the interface representation do not corresponds exactly to the one in the mathematical world. That is to say that the approximated global eigenfunctions do not satisfy relations as (8). However, to have a good representation of the interface in the numerical world, we have introduced some functionnals. Their minimization allows to control the discontinuity of the approximated global eigenfunctions and of their flux.

This methodology has been applied to "2D" and "3D" thermal systems (thermal bridge...). The good results obtained [7, 8] allow from now to study complex industrial problems with modal method.

5 The simulation environment MOTOR

The formerly mentioned ideas of modularity and abstraction were applied for simulation purposes in the MOTOR program. This program takes a description of the split up system at the level of the technical world. For each of the terminal elements, MOTOR looks for a calculation model in its library and translates it into its internal data structure. MOTOR starts then a time transient simulation of the terminal modules and their connections. Each module of the system graph has its own thread of execution in order to parallelize the program and to gain execution speed.

The encapsulation of simulation models into objects keeps all data and rules of a real world object in a single unit throughout the abstraction levels up to the computer representation. Re-use (and sharing) of these independent models is achieved by libraries as done in other environments (e.g. SPARK, IDA, TRNSYS). But it happens quite rarely that one will use exactly the same unit in two different system studies. Thus the aspect of easy modification of the units that widens the possibilities of re-use is considered in the environment. MOTOR facilitates the description of new models as being similar to an existing one through the notion of inheritance.

The main differences of MOTOR from other simulation environments consist in:

- a hierarchical partitioning of the system as opposed to a final global system of equations
- simultaneousness in calculation of the different parts as opposed to serial calculation on traditional computers

5.1 Partitioning in MOTOR

As said before, all information concerning an object, e.g. the thermo-physical properties and laws of the physical world, the equations from the mathematical world, and so on are gathered in a single unit of the computer world.

The cutting off of a system gives us a structure of dependencies between the modules where they must be maintained in the simulation system. It is up to the user to build a pertinent graph because the calculation strategy of MOTOR is based on the hierarchy of the system description. The tree structure is kept in the internal data structure even at runtime of the simulation. Thus, we have computer units (and therefore algorithms to treat) not only in the terminal buds of the tree, but also for every ramification, and this on every level of the tree.

The coupling modules (or composite modules) mainly consist of what we call interfaces in SYMBOL. Generally several interfaces are within one composite module. These interfaces are communication points which connect two or more submodules by its frontiers. Thus, the parent module controls the evolution of its submodules via the interfaces and its own outer frontiers. In Figure 4, the frontier $F_2^{c_1}$ of the component C_1 is connected to the frontier $F_1^{c_2}$ of the component C_2 at the physical level. Besides the connection information, the interfaces contain the binding equations (e.g. equality of temperatures in all submodules and sum of heat fluxes is zero) from the physical level.

When MOTOR builds its internal structure, the frontier informations of the submodules and the interface equations become oriented ones in order to be used to express calculation methods at the numerical level. Besides other algorithmic aspects that are due to parallelization, the user can choose between a general *Newton-Raphson* method and a modified version of the highly optimized MINPACK equation solver [9] for solving the resulting system of nonlinear equations. At each iteration step, some of the coupled variables (e.g the temperatures T_i) are modified to get the residue close to zero (e.g. sum of heat fluxes). A hierarchical system of equation systems is built if the submodules of a composite module are composite modules themselves.

5.2 A natural step towards parallelization

Parallel computers are becoming more available in the scientific community as a means to accelerate time-consuming calculations. Even highly parallelized computers with some thousand processors are emerging. Our approach allows using these facilities if assigning a task to each module (component or interface) of a system. A task is an independant executable piece of code which collects all data but also the active parts concerning an object. Each of such task object can then be assigned to a processor; thus, the simulation code executed on a multiprocessor machine is very close to the "real" system seen from the technical world.

Four different tasks control strategies have been implemented and tested in MOTOR:

- The first method is called the *global synchronized* method. At each splitting level, the equations of all the interfaces and of the responses of the respective coupled modules are gathered in a unique system of equations; the tasks are started synchronously (Figure 8), this means that *either* the parent module *or* the child modules are active at one given time.
- In a second approach called the *single synchronized* method, every interface builds an independent system of equations made of the local responses from the coupled modules. But all

the child modules of each parent module are still started synchronously. (Figure 8).

- In the *asynchronous* method, every interface and module evoluates independently. Only at the end of each time step, when all interface tasks have signaled a stable solution, the parent module resumes activity and passes the results to its own parent module (Figure 9).
- Finally a fourth method has been implemented: the user may specify the order of execution. This *serial* method inhibits all simultaneousness and is used for chained calculation.

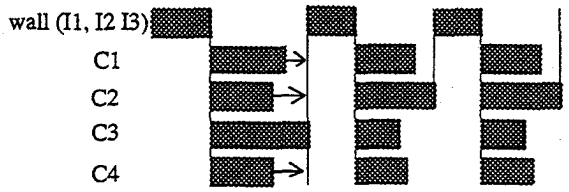

Figure 8: Activation of tasks in the synchroneous strategy

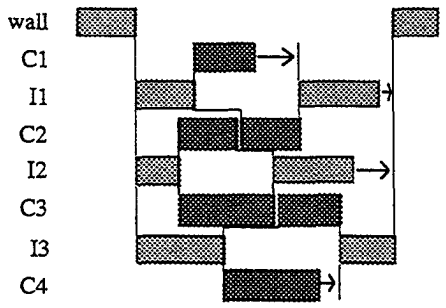

Figure 9: Activation of tasks in the asynchroneous strategy

When the interfaces are treated as independent systems of equations, one has to be aware of the numerical problems that may arise. In this case, a child module does not always see the same input values at all its frontiers when it is activated over a given time step. This is true above all in the asynchronous case and it sometimes leads to a few supplementary iterations. On the other hand, we treat matrixes which are much smaller than for the global system of equations, which compensates the eventual supplementary iterations and module evaluations. On the positive side, we achieve at the same time a higher degree of flexibility; we can for example implement different time steps per module without increasing the complexity of the solver kernel.

The work was done on a single processor workstation. The simultaneousness of the different threads of execution can easily be simulated itself by an Ada-runtime (MOTOR itself is currently written in Ada) or the evolving POSIX standard or SUN's lightweight-process library (both in C). A big performance gain can be expected when running MOTOR on a real parallel computer; porting the MOTOR code to this kind of machine should not raise problems since it already was one of the major design goals of the program. Most probably, today's users still have single processor machines. In this case the simulation program can easily be reconfigured to not use the tasking capabilities and to avoid the (more or less) expensive overhead to simulate the simultaneousness and to get a performant simulation program for today's computers.

5.3 Example: a Room

The flexibility of the simulation environment can be shown with this example in which we use models of very different origins and a multi-level hierarchical structure. In order to evaluate the thermal comfort, we modelled a room with internal radiation, convection and heat conduction, and excitated by outside temperature and solar radiation (Figure 10).

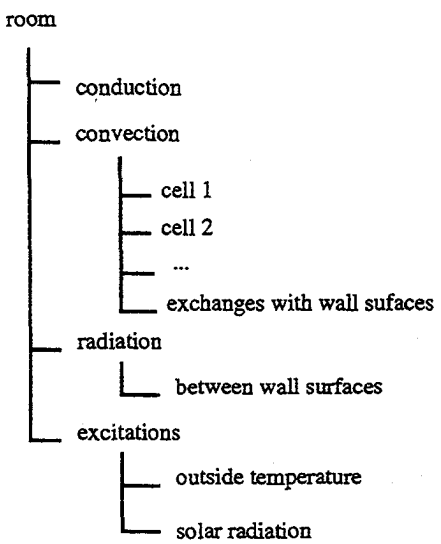

Figure 10: Split up of a room

The first split level correponds to separate the system in four parts; the first one contains the conduction in the walls, the second one the radiative exchanges between the wall surfaces, the third one the convective exchange inside the air volume, and finally the fourth one contains the modules which represent the excitations of the systems. Heat conduction in the surrounding walls has been modeled using reduced modal models. Natural convection in the room has been modelled using 3x3x3 small subcells exchanging air and verifying the equations proposed in [2]. The outputs of the model are the air temperature and the radiant temperature in each cell.

Results are not presented here because they will never show the advantages of this kind of approach; they are only obvious during the modeling process itself (e.g. when designing, writing, coding the model), or when doing intensive repetitive simulation work. Nevertheless, some results of simulations realised with this model are described in [15].

6 Conclusion

Modularity has been a strong leitmotiv during all the design of the SYMBOL environment. We showed that the resulting environment presents some interesting advantages, but it has been necessary to make an abstraction effort in order to be able to describe the modeling process under a coherent form, pertinent for different concrete purposes such as modal synthesis or object oriented simulation. One must remind that the arbitrariness of the splitting is an unsolved problem. Some elements relative to the influence of its pertinence is treated in [14]. This structural strategy is finally intellectually satisfying because it looks very close to our intuitive representation of the "real" world. In other respects, this strategy allows to save effort and time, and leads users to test more configurations.

References

[1] Blanc Sommereux I., 1991 *Etude du couplage dynamique de composants de bâtiment par synthese modale*, These de doctorat, Ecole des Mines de Paris

[2] Bouia H. and Dalicieux P., 1991, "Simplified Modeling of Air Movements Inside Dwelling Room", *Building Simulation'91 Conference Proceedings*, NICE

[3] Bourquin F., 1989, "Component mode synthesis of second order elliptic operators", *Comptes rendus de l'Academie des Sciences*, tome 309, Serie I, 919–922, PARIS

[4] Dubois A.M. et al., 1991, "The model coupling problem: methods used in some building analysis tools and the ALMETH propositions", *IBPSA Conference Proceedings*, Sophia-Antipolis

[5] El Khoury K. and Neveu A., 1989, "Analyse modale des systemes thermiques en presence de transferts non reciproques", *International Journal of Heat & Mass Transfer*, **32**, 2, 213–226

[6] Flament B., Blanc Sommereux I. and Neveu A., 1991, "A new technique for thermal modelling of building: the modal synthesis", *IBPSA Conference Proceedings*, SOPHIA - ANTIPOLIS

[7] Flament B., Bourquin F. and Neveu A., 1993, "Synthese modale : une methode de sous-structuration dynamique pour la modelisation de systemes thermiques complexes", *International Journal of Heat & Mass Transfer*, (to be published)

[8] Flament B., 1993, *Synthese Modale de Systemes Thermiques*, These de doctorat, Ecole des Mines de Paris

[9] Garbow B.S., Hillstrom K.E. and More J.J., 1980, *The MINPACK Project*, Argonne National Laboratory (March)

[10] Lefebvre G., 1987, *Analyse et reduction modale de modeles de comportement thermique de bâtiments*, These de doctorat, Universite Pierre et Marie Curie, PARIS

[11] Lefebvre G. et al., 1990, "Applying the modal method to thermal modeling", *International Heat Transfer Conference Proceedings*, 3, JERUSALEM

[12] Litz L., 1981, "Order reduction of linear state models via optimal approximation of the non dominant modes", *Large Scale System*, North-Holland Publishing Company, 2, 171–184

[13] Michaleisco G.P., 1979, *Approximation des systemes complexes par des modeles de dimension reduite*, These de Doctorat d'Etat, PARIS

[14] Nataf J.M. and Ebert R., 1993, "On the Efficiency of Partitioning in Object-Based Simulation". *Building Simulation'93 Conference Proceedings*, ADELAIDE

[15] Nataf J.M. and Wurtz E., 1993, "Application of the SPARK environment to 3D air flow problems", *Building Simulation'93 Conference Proceedings*, ADELAIDE

[16] Oulefki A. and Neveu A., 1993 "Reduction par amalgame modal d'un modele thermique", *Journal de physique*, III, PARIS

Aknowledgements

The work presented in this paper has been realised with the financial support of the french "Agence De l'Environnement et de la Maitrise de l'Energie.