A graphical man machine interface for modular HVAC system simulation programs.

*H Riaux, **M Molnar, *J Miriel, *P Boinet
*Laboratoire GTMa de l'INSA de RENNES
**Laboratoire informatique de l'INSA de RENNES
20 av des buttes de Coesmes 35043 Rennes Cedex France

ABSTRACT

The presentation of most HVAC simulation programs are out of date and batch processing type. Their use is complex and often requires a good data processing knowledge. Moreover, the data entry is usually long and the search for conceiving or typing mistakes is not easy.

This present work shows how to update the simulation environment and how to satisfy the modem work station requirements by using a graphical manmachine interface.

The proposal software CAMOSIM is developed through a modular system HVAC simulation program "TRNSYS ",using "a workstation under UNIX, with X-Windows and Athena widgets". This program is composed of two main modules : the first one enables to generate global model libraries and the other one specific applications.

GLOSSARY

Component model: a model describing a virtual physic component mathematically.
Real component : a component model with its own values
Connections : connections between real components
Application : the set of the real components and the connections in a real case of simulation
Model library : the set of component models.
Catalog : a real component library .

1. INTRODUCTION

1.1 Advantages of the simulation programs

The development of computers has allowed to define many simulation programs, such as HVAC system simulation. The handling and the different behaviour between the components of the system are described by means of more or less sophisticated mathematical models. These simulation programs allow to estimate the global system performance and to study the different solutions without having to realize them in real size . Therefore, they represent a valuable help for designing.

1.2 Difficulties with present simulation software:

Most current simulation programs are out of date and batch processing type. They have to be collected within required data for simulation in data files whose structures are fixed.

Most of the time, these files are very tedious to enter (high number of parameters) and require rigour.

Checking their validity is extremely delicate and finding typing mistakes is not easy.

1.3 Efficiency of simulation programs

Simulation methods are numerous and have been the object of experimental validations. Computers have higher and higher computation capacities. For the time being, the efficiency of simulation programs depends more on entering facilities or on parameters modifications than on the speed of calculation.

1.4 Modern workstation requirements

A modern simulation program should be able to offer a certain number of functionnalities related to connect man - machine:

- a guidance for the user through the proposition of various choices at every level of the process of creation of the application.
- a modular structure permitting some modifications and further developments.
- an easier entering of the parameters of the application components. (proposition of default values, units verification).
- a visualization of the existing connections between the different modules (inputs from outputs).
- a control of the coherence of information exchanged between the connected modules.
- the real time control of the computation process by visualizing some parameters or selected signals.
- the modification during the computation process of the module parameters.
- the possibility to stop the computation process before its completion.
- the automatic or optional storage for a particular application including:
  * the set of component,
  * the information collected during the computation,
  * the final results.
- the beginning of a new work session through a duplication and modification of an existing application.

1.5 CAMOSIM
(Creation of Application for a MOdular SIMulation program)

CAMOSIM offers to answer the demands of a modern workstation, except simulation in real time. For that purpose, it uses modern tools such as X-Windows in order to facilitate man machine communications. Moreover, CAMOSIM automatically generates the subsequent data files of the modular simulation programs.

2. A MODULAR SYSTEM SIMULATION PROGRAM : TRNSYS

2.1 introduction

TRNSYS is a transient system simulation program with a modular structure, developed by the Solar Energy Laboratory of the University Of Wisconsin. Initially, TRNSYS was created for solving active solar systems and for calculating building loads. Nowadays, it is extended for simulating HVAC systems in buildings. The modular nature of TRNSYS gives the program tremendous flexibility, and facilitates the addition to the program of mathematical models not included in the standard TRNSYS library.

TRNSYS is composed of approximately 80 subroutines, which are organized into three groups :

- **Frontend modules** processes and orders the information contained in the simulation deck.
- **Support modules** which are called by certain components models, like DATA files or DIFFEQ called to solve differential equations.
- **component models** about HVAC systems (pump, heat exchanger ...), about building components and also thermal controllers.

A system is defined to be a set of components, interconnected in such a manner as to accomplish a specified task.

2.2 description of a component model

The TRNSYS library includes many of the components in thermal energy systems. Every component subroutine describes mathematically the performance of the component, and must be defined by :

- Parameters,
- Inputs,
- Derivatives (time dependent variables).

the outputs of a component model are connected to the inputs of another one, as shown in figure 01.

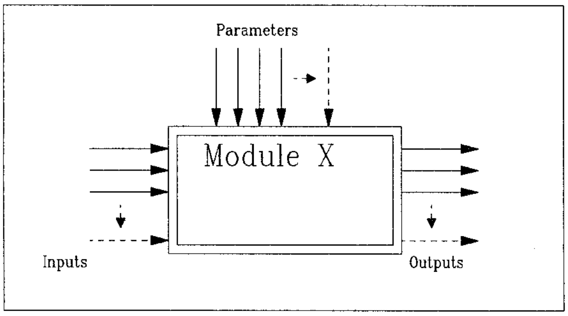

figure 01 : Component model

Many components may operate in any of several modes, offering differing degrees of model complexity. For example, a solar collector may be described in four modes. The parameters values and the input initial values must be given with units recommended for TRNSYS. For example the energy flux unit is kJ/(m^2 * hr).

2.3 information flow diagrams

an example : a solar water heating system :

In this example (figure 02), a water solar collector (type 1) is connected to a water storage tank (type 4). This system with an auxiliary heater is designed to supply the hot water domestic needs.

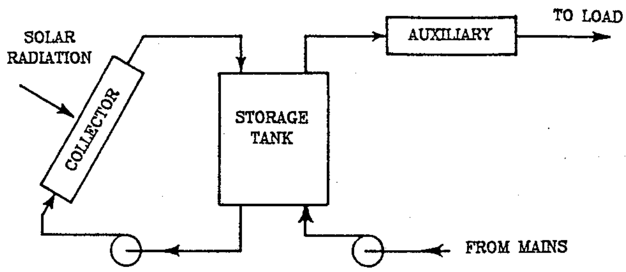

figure 02 : A solar water heating system

For simulation, TRNSYS requires files of meteorological data at short and regular time intervals (type 9). In this case, a solar radiation processor is included in the information flow diagram in order to convert the radiation data into a form useable by type 1 (figure 03)

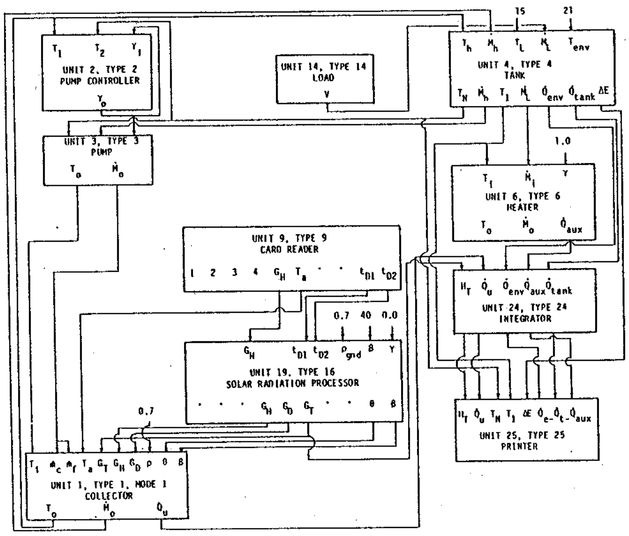

figure 03 : information flow diagram

3. GENERALIZATION IN THE STRUCTURE OF THE MODULAR SIMULATION PROGRAMS

3.1 topology of component model

Taking the set of the real components used in a lot of applications of simulation of HVAC systems in building, we can gather components having the same mathematical model.

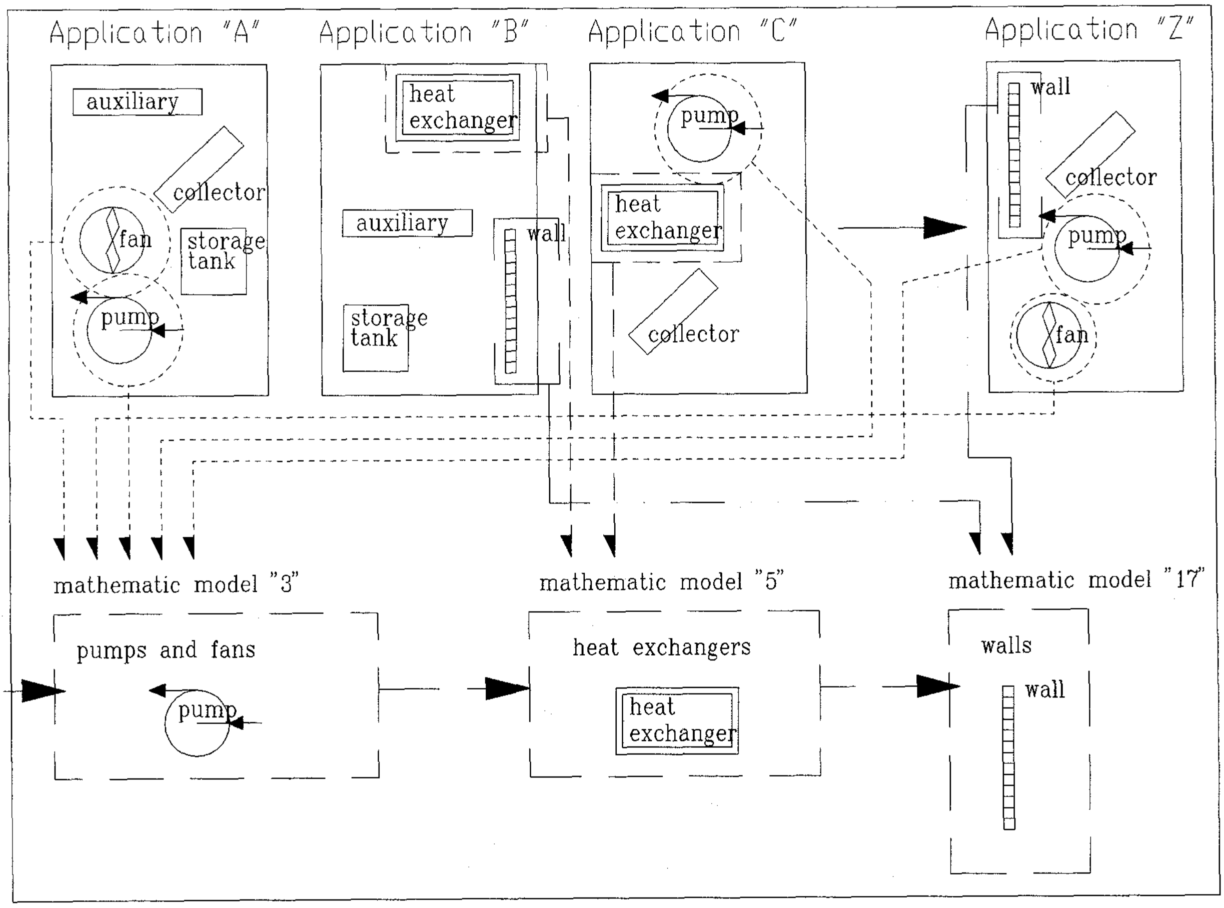

figure 04 : first gathering of components

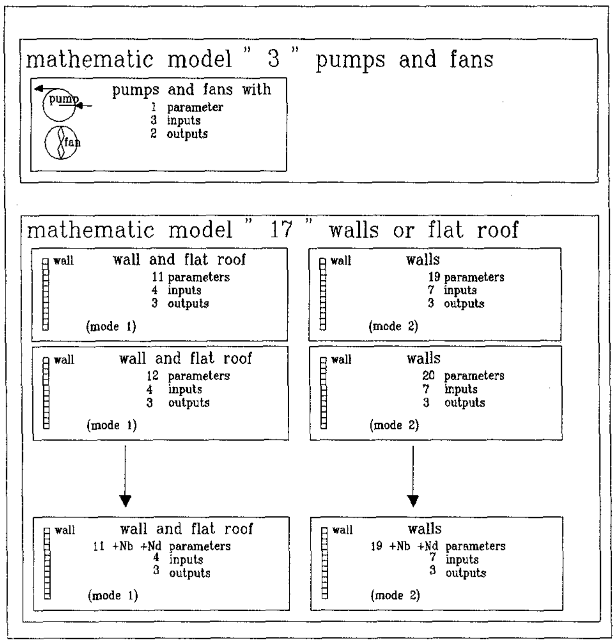

figure 05 : second gathering of components

Within each group of components having the same mathematical model of simulation , we gather those possessing an identical number of elementary data (same number of parameters, same number of inputs, same number of outputs, knowing that the values of these data can be different).

The set of these common characteristics makes up what we call a component model.

its structure is composed as follows:

* name of the component model
* type of the component
* parameter number 1: name, type, default value
* ....
* parameter number k: name, type, default value
* input number 1: name, type
* ....
* input number m: name, type
* output number 1: name, type
* ....
* output number n: name, type

The component models are grouped into model libraries of two types :

- a global model library containing almost all of the component models.
- Some particular model libraries made for specific applications using only one part of the component models, reserved to some users, for certain specific applications.

3.2 Creation of a specific application

Through one selection in one of the different model libraries, it is possible to create a real component used for a simulation application by fixing the values of its parameters and by defining its connections.

This operation shall be applied for each required application component.

In the applications, the structure of the basic data comes as follow :

* name of the real component
* name of the component model
* parameter number 1 : name, value.
* ....
* parameter number k : name, value
* input number 1 : name, origin (name and characteristics of the output), initial value
* ....
* input number m : name, origin (name and characteristics of the output), initial value

This subroutine gathers the informations necessary to design an application and necessary as well to the future simulation.

At this level the connections definition takes place, and coherence controls are made (size, units).

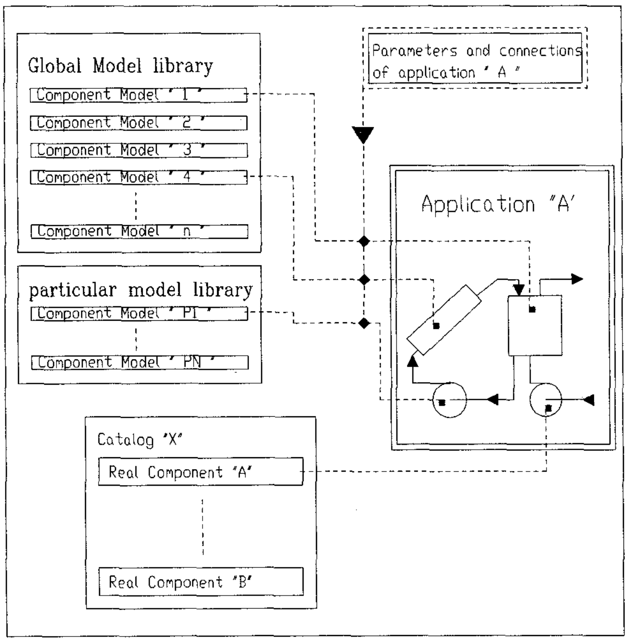

figure 06 : creation of an application

The set of the application real components is entered in a data file within a suitable structure with the modular simulation program TRNSYS.

4. CHOICE OF THE DEVELOPMENT ENVIRONMENT

It is necessary to have a graphic tool permitting the man-machine dialogue and offering hierarchy menus and windows.

Using the mouse is necessary, as a pointing and quick selection tool, interfaced with direct selections on stockers, on the screen or in menus. To ensure possibilities of improvement and further complements, as well as the portability, we have to use standard products, in the public domain, able to support a object oriented programming. For its rational use, the solution must be multi-users and multitasks.

According to these elements, the group of development has decided to choose the X-Windows environment under UNIX, the use of C-language and the Athena widgets.

5. CAMOSIM A MODERN GRAPHICAL INTERFACE

5.1 Generalities

CAMOSIM conception is in line with the generalization proceeding dealt with in the third paragraph. When building up CAMOSIM interface modules, the particularities of the TRNSYS simulation language have been used. But with some modifications, CAMOSIM can be used again to interface other modular simulation languages. Find hereafter CAMOSIM functionalities through TRNSYS.

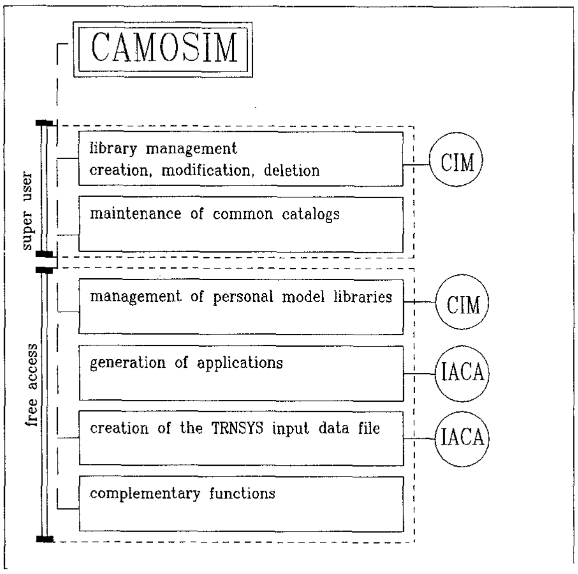

figure 07 : functionalities of CAMOSIM

As it can be noticed, CAMOSIM includes two main modules :

*CIM (Maintenance of model libraries) which allows to create, to modify and to delete the common or personified model libraries. Naturally, the management of the common libraries is led by a data-base super user.

Finally, CIM assures the management of component models at any step.

*IACA (application creation) which permits to parameter the objects specific to an application and to generate the compatible file with the modular simulator.

5.2 General principles

CAMOSIM structure relies on four general principles:

5.2.1 Global model library and catalog conception

From mathematical model classes of a modular simulator language, we can build all the possible component models and collect them in a global model library which gives all the basic elements of this simulator language.

To get better approach to the component models, this library can be divided into large families. The creation of pre-established object catalogs with their parameters but without connections is essential in the parallel direction with the component models . Then, the common catalogs include the real components with theirs parameter values but without connections.

5.2.2 Particular model library conception

A specific application requires only a part of the global model library. Using it during a simulation is very hard. Thanks to CIM, model libraries can be created by application or according to the needs of the users, under number restriction.

CIM module ensures the selection of different component models from different models and makes copies of component libraries easier.

5.2.3 Application creation

In order to create a concrete application, the real components and their connections are determined from one or several model libraries (which contain the list of the component models in regard to the created case). A similar proceeding can be followed by using the real components which appear in the common catalogs.

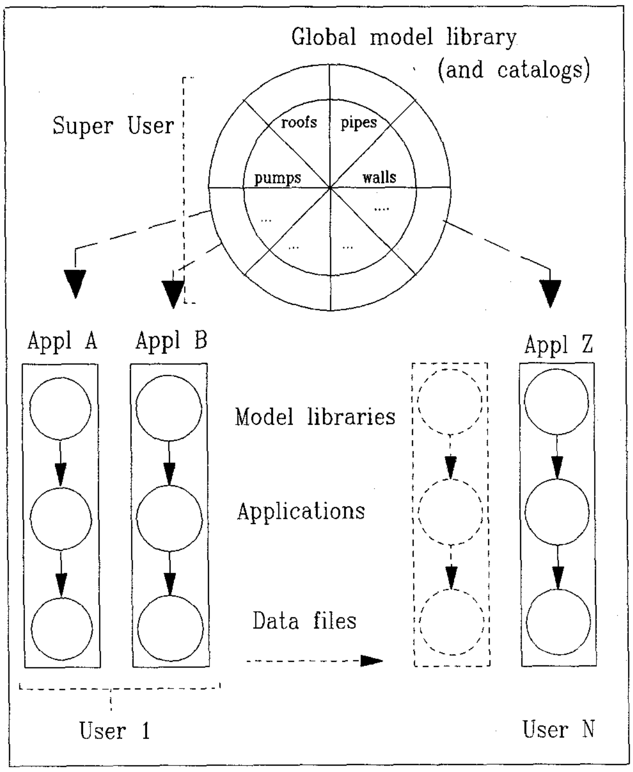

figure 08 : general principles of CAMOSIM

5.2.4 Task distribution

Getting access to certain functionalities is not possible. For example, on the global model library, the interventions will be reserved to the data-base super users only. Once on his working place, the user will have access to his own files. The data will be protected with the rights of UNIX access and with CAMOSIM software.

5.3 C.I.M.

With the help of the first module, we can either build up a global model library from modules coming from TRNSYS or create, modify and improve a model library for a special application, using for example a partial recopy of the global model library.

This creation proceeding and keeping model libraries up to date will need a series of arborescent windows offering several choices.

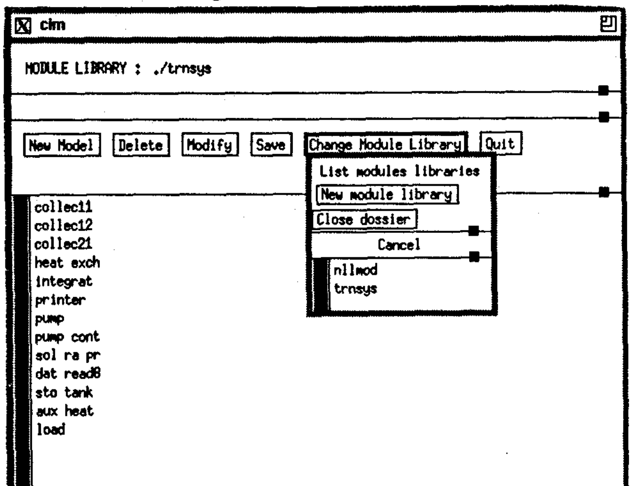

figure 09 : CIM main window

The main window will give us the possibility to choose between several functions :

- searching of the working model library
- saving of the active model library,
- going back to the operating system, considering or not the last modifications.

On the selected model library, it will be possible to:

- create a new component model and add it to the previous list,
- delete a selected component model,
- modify one or several data of the selected component model.

This window has a notice line of possible error messages or request confirmation lines about the active subject.

The functionalities such as modifying, deleting, inserting , are only active when the global model library is selected. Only the super user of the common data-base management can do what he wants about them.

All the users can consult their own files and all the global files whose access is permitted by the super user.

The selection of the function "change model library" give us the possibility either to select in an exhaustive list, the working model library or the model library to be deleted or to create a new one.

When we create or modify a component model, the activation of the corresponding functions displays a sub-window (figure n° 09), with the following functions:

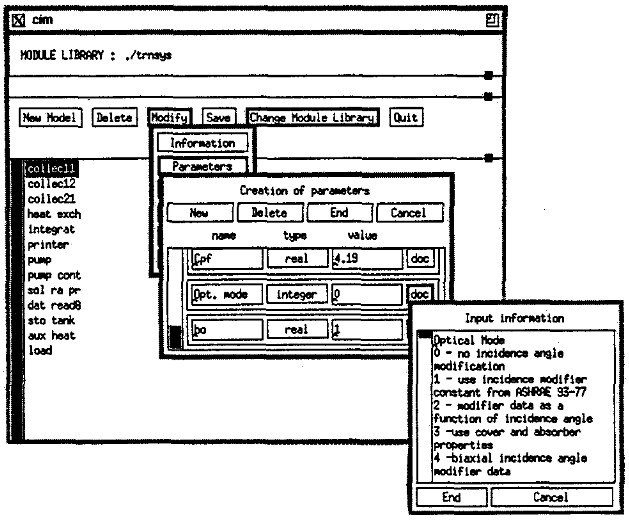

figure 10 : example of CIM sub-window

- access possibility, through an information window, to information about the component model which can be called whenever you want it, with a help instruction.

- creation of obligatory entities : parameters, inputs, outputs, module TRNSYS issues.

- information validation and save under a model component name.

- deletion of an obsolete component model.

- duplication of a component model for creating another without having to reinitialize them, if the information keep very similar (for instance, heat exchanger or solar collector duplication with a different mode).

- the input, output and parameter selection keys label, will allow us, as seen before, to create a new entity, to delete it, to exit, saving it or not. For a data, it's possible to reference it for a further check, to give it a type (integer, real, string) and to add a proper documentation to it.

For the parameters, it's possible to enter a default value.

5.4 I.A.C.A.

The second basic subroutine is built with the same principles as CIM, but it starts from the model library to generate an application.The data zone and the connections in order to define the set of real components must be completed.

In the main window,the following functionalities are displayed:

- selecting an application in a exhaustive list in order to use it in a work session, to destroy it or to create a new application.

- selecting the global model library or the particular model library in order to select the exact component model.

- creation and addition of a new real component.

- deleting or modifying of a real component.

- saving the modified data.

- escaping with or without save.

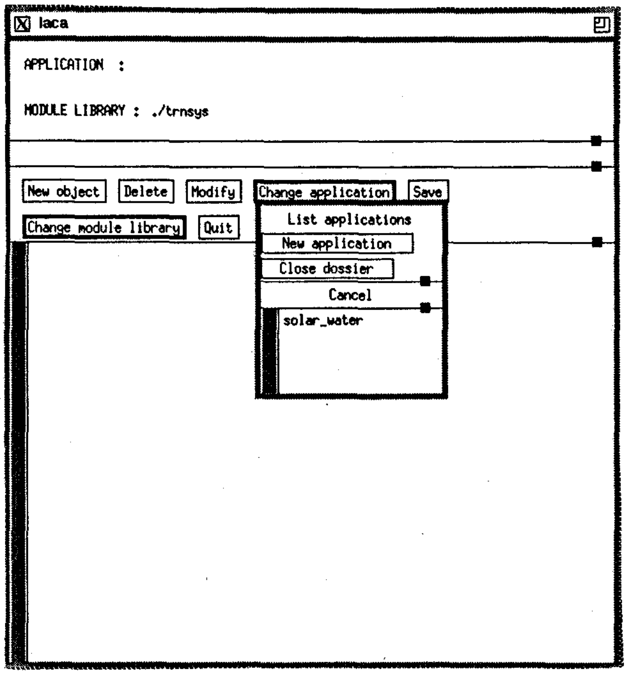

figure 11 : IACA main window

The modify functions, as CIM, are only available in the own user directory.

A component model is selected in order to create or to modify a real component (figure 12), the information window can be displayed, then the other functionalities can be used:

- creating information.

- entering parameters with one word for the validity control and one value for the simulation.

- defining connections (input from output). The inputs require a word for the validity control and one initial value.

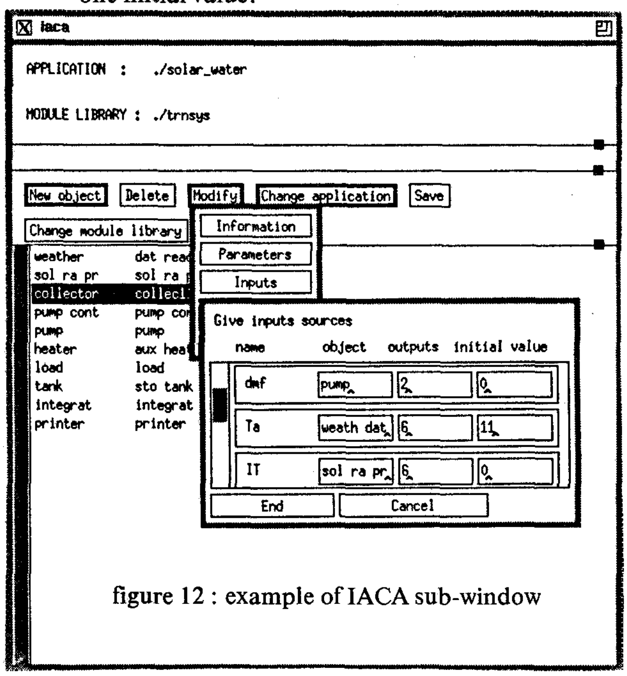

figure 12 : example of IACA sub-window

During the entering of data , the IACA subroutine controls the connections between each application component.

When the application is enabled, the subroutine generates automatically a new file with the required format for the simulation. The running of the simulation program is optional

```
SIMULATION        0.000E+00    1.680E+02    2.500E-01

WIDTH    72

UNIT  9     TYPE  9      CARD READER
      PARAMETERS   8
      6.000E+00    1.000E+00    -5.000E+00    4.187E+01    0.000E+00
      6.000E+00    5.556E-01    -1.778E+01

UNIT 16     TYPE 16      SOLAR RADIATION PROCESSOR
      PARAMETERS   6
      3.000E+00    1.000E+00    8.000E+00     4.000E+01    4.871E+03
      0.000E+00
      INPUTS  6
       9, 5         9,19         9,20          0, 0         0, 0
       0, 0
      0.000E+00    0.000E+00    0.000E+00     7.000E-01    4.000E+01
      0.000E+00

UNIT  1     TYPE  1      COLLECTOR
      PARAMETERS  12
      1.000E+00    1.000E+00    6.500E+00     4.190E+00    1.000E+00
      5.000E+01    7.000E-01    1.500E+01    -1.000E+00    4.190E+00
      1.000E+00    1.000E-01
      INPUTS 10
       3, 1         3, 2         3, 2          9, 6        16, 6
      16, 4        16, 5         0, 0         16, 9        16,10
      6.000E+01    0.000E+00    0.000E+00     1.100E+00    0.000E+00
      0.000E+00    0.000E+00    7.000E-01     0.000E+00    4.000E+01

UNIT  2     TYPE  2      PUMP CONTROLLER
      PARAMETERS   3
      3.000E+00    1.000E+01    1.000E+00
      INPUTS  3
       1, 1         4, 1         2, 1
      1.500E+01    6.000E+01    0.000E+00
```

figure 13 : example of TRNSYS input file

6. CONCLUSIONS

Through the presented application, we intended to show the advantages obtained by a man-machine interface, in the modular simulation systems.

- entry of data and use for non specialists are easier.
- control at any step of the data base creation and applications are achieved.
- creation of a new application from the specific library is faster.

This work can be extended to the whole TRNSYS library by adding complementary functionalities (management and modification of the applications, graphic display of the flow diagram, etc...). This development could update modular simulation programs (HVAC SIM+, ASTEC, CSMP, etc..). These programs have been validated, but the interactive qualities are very poor.

REFERENCES:

BOOK :

Dumeur, J.A. Le Borgne, M. Bassini : "PratiX utiliser et programmer X", Editions C2V.

MANUALS :

Gettys, R.W. Scherfler, R. Newman : "X Lib - C Language X Interface".

McCormack, P. Asente, R.R. Swick : "X Toolkit Intrinsics - C Language X Interface".

R.R. Swick, T. Weissman : "X Toolkit Athena Widgets - C Language X Interface".

Solar Energy Laboratory : "a TRaNsient SYStem Simulation program", University of Wisconsin - Madison.

RESEARCH REPORT :

P. Anne, V. Gateau, J.F. Germain : "Création d'une interface graphique IACA et CIM", INSA Rennes, Janvier 1990.