

# COMIS 3.0

## A NEW SIMULATION ENVIRONMENT FOR MULTIZONE AIR FLOW AND POLLUTANT TRANSPORT MODELLING

Dr. Roger Y. Pelletret  
Head of Software Development Division  
Leader of IEA/ECB/Annex 23-Subtask 1

Dr. Werner P. Keilholz  
ISE Project Manager  
Software Development Division

CSTB (The French Building Research Centre)  
BP 209 - 06 904 Sophia Antipolis - France  
e-mail: pelletret@cstb.fr

### ABSTRACT

COMIS 3.0 is a new simulation environment developed in the framework of IEA/ECB Annex 23. To our knowledge this is the first time that one of the Annexes of IEA/ECB produces as its main result a validated software not only designed for research laboratories but also intended to be used by engineering and consulting companies.

Annex 23-subtask 1 aimed at developing a multizone air flow modelling system (COMIS<sup>1</sup>) encapsulated in a Simulation Environment, in order to facilitate access to, and use of COMIS.

In this paper, the main features of this new multizone airflow and pollutant transport simulation code are presented. Examples of applications to real cases are given.

### KEYWORDS

Multizone, Airflow, Modelling, Simulation Environment.

### INTRODUCTION

At IBPSA'93 CSTB's ISE<sup>2</sup> R&D program was first introduced. The main objective of this R&D program is to transfer innovative simulation tools to practitioners. To achieve this goal, a generic Simulation Environment, called IISiBat<sup>3</sup>, has been developed. This object oriented simulation environment embeds many features such as structured libraries of models, a comprehensive on-line model documentation, macro modelling functions, a multi-run function, etc.

The generic concepts underlying the ISE project have been applied to TRNSYS and, at IBPSA'95, the

first operational application of the ISE project, TRNSYS 14.2 w. IISiBat, was presented.

At the same time, IEA/ECB Annex 23 was working on the development of a new "air flow and pollutant transport in multizone buildings" simulation software. This software was based on the first COMIS model: COMIS 1.0. COMIS 1.0, a Bernoulli equation based program, has been developed in the late eighties at LBL by an international team.

The specific objectives of Annex 23/Subtask 1 were:

- to develop a new, validated, program to compute air flow and pollutant transport in multizone buildings;
- to facilitate the use of this program in order to ease its dissemination.

To reach the first objective, new modules have been added to COMIS so that the final version (COMIS 3.0) can compute (i) pressures and air flows in HVAC networks, (ii) pressure distributions on building envelopes and in internal zones and (iii) pollutant concentrations in multizone buildings. Some of the new modules are the implementation of R&D work carried out in previous annexes. This is the case for the inhabitant behaviour module, the large vertical opening module and the single sided ventilation modules; these three modules were Annex 20 results.

To facilitate the use of COMIS, the IISiBat Simulation Environment has been used to encapsulate COMIS. Apart from this main development, COMIS is also available as a stand alone Fortran program and there is also a DOS program front-end (COMERL) developed by, and available through EMPA (Switzerland) [5].

<sup>1</sup> Conjunction of Multizone Infiltration Specialists

<sup>2</sup> Intelligent Simulation Environment

<sup>3</sup> Intelligent Interfaces for Simulating Buildings (i.e. Batiments)

## Software Architecture Summary

In this paper, we can not describe the software architecture of IISiBat in detail. Readers who are interested in getting more information can refer to [7] and [9]. We will here focus on the functionalities of the COMIS simulation environment from a user point of view. However, a short summary of the software architecture will be given here. This should enable the reader to compare our approach to other approaches to the simulation environment problem.

IISiBat is based on a *layered* architecture. The implementation consists of three *layers* [7]:

### 1. The kernel layer

This is the 'heart' of the tool. The objects defined in this layer are the basic bricks used by the system to build the simulation environment for a given simulation tool. This layer also contains some general-purpose tools, such as a text editor.

### 2. The simulator specific layer

In this layer, objects from the kernel layer are redefined to account for specific features of a given simulation tool. For example, we define here how a COMIS input file (*CIF*) is generated from the internal, *generic data representation* of a simulation project. This is done by redefining the 'write-input-file' message of the `comis-project` object which is based upon the `generic project` object.

### 3. The user layer

In this layer, the behavior of specific models can be redefined. This is not only possible for standard models, but also for all user-defined models, because the system automatically generates class definitions for these objects, thus allowing for method redefinition.

Developers can redefine methods for any IISiBat object and thus influence its behavior. A typical example is the implementation of so-called 'attachments' in IISiBat for COMIS. Attachments are used to define schedules attached to objects, such as temperature schedules attached to zones or opening angle schedules attached to windows. This concept does not exist in the generic IISiBat tool. To allow the user to attach schedules, we decided that he/she should be able to click on the lower part of the graphical representation of an object, to access a specialized schedule editor. To be able to implement the desired new behavior it was sufficient to redefine a method called `make-sensitive`, which is

defined in the kernel layer. This method defines how the object reacts to clicks. The redefinition of the method can simply call code implementing the desired new behavior.

Many other methods are defined for IISiBat models, forming an API (application programmer's interface). Their redefinition allows us to control many aspects of the model's behavior, ranging from its graphical representation to the way it is represented in the simulation tool's input file (the *CIF* in the case of COMIS).

In contrast to other "simulation environment projects", such as the NMF-based IDA project [10], IISiBat does not attempt to provide a complete environment from scratch, including a numerical methods etc. It rather 'wraps' itself around existing software, such as COMIS, and encapsulates this software in a convenient simulation environment. Within this architecture, certain advantages of the object oriented concept of the implementation still penetrate to the end user. On one hand, the system can easily evolve, taking new requirements into consideration and be adapted to requirements emerging during the use of the software. On the other hand the user himself is invited to organize numerical models encapsulated with IISiBat in an object oriented manner (using the library manager and the libraries), although the underlying simulation code may be implemented in a language like FORTRAN. Of course the latter features would be most useful if IISiBat were one day applied to a NMF based simulator.

## Application to COMIS

COMIS 3.0 is a package that includes the FORTRAN simulation code and a specialized version of the IISiBat simulation environment. We will now describe this implementation from a user's point of view.

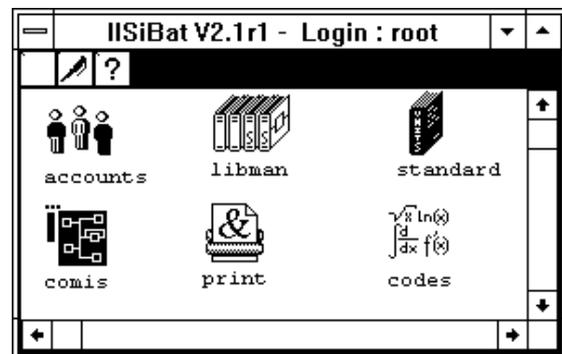


Fig. 1: COMIS 3.0 "Root window".

The root window contains six applications but other modules like a spreadsheet could be easily added. User account manager: one of the major problems that limit the spreading of so-called Building Performance Evaluators (BPEs) as COMIS is the fact that the same simulation environment is usually supposed to be sufficient to deal with various aspects of the modelling / simulating process. This would mean that users that do substantially different tasks such as model development, data input, system modelling, result interpretation, etc., would be supposed to, at least, have the same needs and problems. COMIS 3.0 distinguishes different types of users and offers each of them adapted front-ends and functionalities. We will not go deeper into these details and in the following presentation we assume that the user is a so-called "model developer".

“Libman” is the COMIS library manager. This is the place where all the COMIS modules and projects are stored (see hereafter).

“Standard” is a dictionary of units very useful to allow one to use any unit he (or she) wants when modelling and simulating; thanks to the unit dictionary, unit conversions are automated in COMIS 3.0 when inputting data.

“Print” is a utility program which permits to print screens. “Codes” is an other utility program to tell IISiBat where to look for the simulation codes (e.g. different versions of COMIS can be stored on the computer hard-disk).

COMIS is the main application. A click on this icon opens an “Assembly panel” which is the main user interface to COMIS.

### Library Manager

In COMIS 3.0 the components and the simulation projects are stored in object oriented libraries (see Fig. 2).

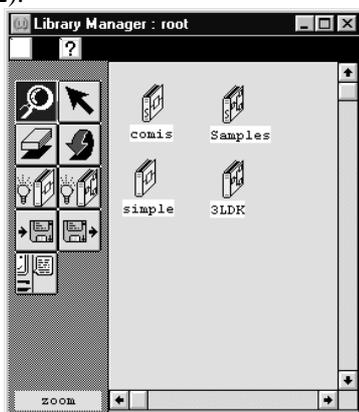


Fig. 2: COMIS 3.0 library manager.

The tools allow to create new libraries of either models or projects, to export or import libraries and to recompile COMIS in case the source code has been modified.

The COMIS “book” contains the COMIS standard components (see Fig. 3). Another book (Samples) contains some pre-defined projects, ready for simulation (the user tests which have been used in Annex 23 to validate COMIS).

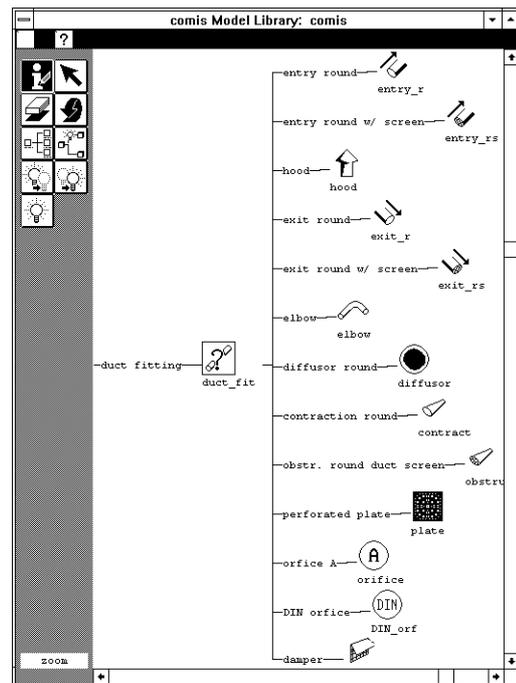


Fig. 3a: COMIS 3.0 standard component library.

The components are arranged in a tree. The nodes of this tree represent “classes”. Object oriented programming has one directly perceptible advantage for the end-user: as we can see in fig. 3, there is a node called “duct\_fit” and all the duct fittings are attached to this node. General properties of duct fittings are defined at the node level; the simple fact of graphically attaching a component to this node makes the component inherit all the node’s properties. This is very helpful to be efficient when developing large libraries of models.

In addition, the documentation for each component is extensive in COMIS 3.0. The standard documentation format called Proforma is used. Future versions of the software may use NMF (Neutral Model Format).

In this version, a new, simplified component library will also be available. The generic components available in COMIS, such as *zone*, *link* or *facade* proved to be too elementary for most users. We found it would be preferable if the user could reason

in terms of *interior wall*, *exterior wall*, *ceiling*, *floor*, etc., instead. Therefore, a simplified model library has been created from the basic, standard model library. This has been done by either adapting the elementary components to specific functions (e.g. a crack can be specialised to represent a wall) or by combining existing components to more complex ones (e.g. a *facade element* always comes with an *external node*). To create this library, extensive use of both the inheritance and macro mechanisms of the IISiBat environment has been made.

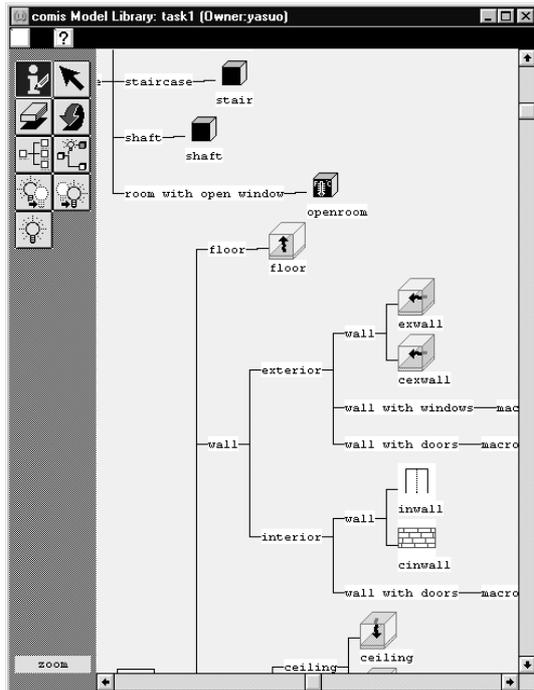


Fig. 3b: COMIS 3.0 simplified component library.

### Assembly window

A COMIS project is constructed by assembling components in the COMIS assembly panel (see Fig. 4). One of the tricky ways of modelling complex systems with COMIS 3.0 is to use the concept of “macro-component”. For example, the ten-storey building displayed on figure 4 is composed of ten macro-components plus a macro-component for the outside conditions; each of the ten storeys is in fact an assembly of COMIS modules as shown in figure 5. In other words, each storey is a different instantiation of the same macro-component.

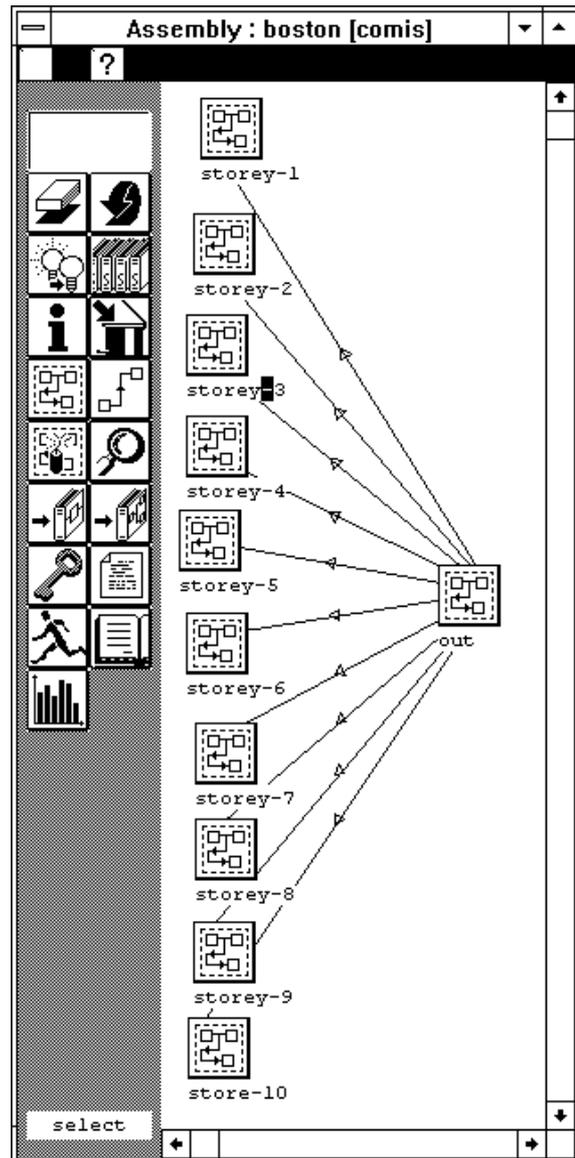


Fig. 4: Modelling of a ten-storey building.

This is a very important feature of COMIS 3.0. The “macro-component” functionality allows one to add to the COMIS standard library, modules (i.e. macro-components) that are often reused in Multizone Building Air Flow modelling.

The content of the macro-component “storey” is displayed on figure 5.

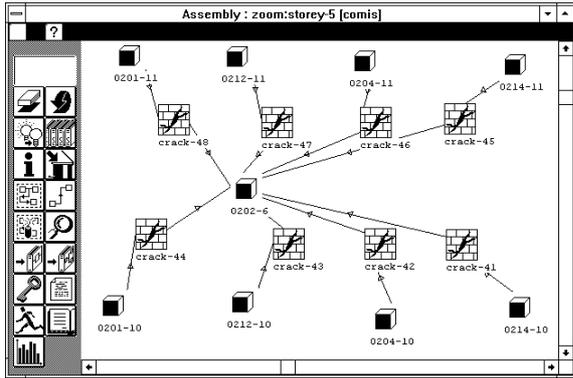


Fig. 5: “Storey”, example of a macro-component.

To complete the system modelling, parameters have to be valued. Entering the model parameter values is very simple: a click on the upper part of the model icon will display the parameter window (see Fig. 6)

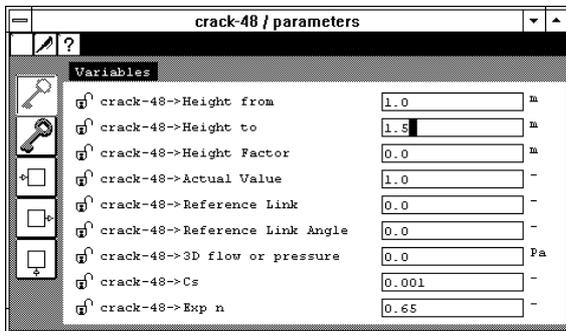


Fig. 6: Model parameter window.

Once the project is ready for simulation, a simple click on the “run” button launches the interpreter that verifies and translates the scheme into a COMIS Input File and launches the simulation. Results can be displayed in a separate window thanks to the “edit result files” button.

### FIELDS COVERED BY COMIS 3.0

Many modules are embedded into COMIS (see [2]) from Air Flow Components such as cracks, test data components, windows, doors, vertical apertures (2-way flow), ducts and duct fittings, fans, flow controllers to pollutant sources and occupants, plus the possibility of defining schedules attached to most of the components (i.e. closing and opening of windows in relation to inhabitant behaviour). All these modules allow various applications such as:

- sizing of mechanical ventilation system
- effects of retrofitting measures on ventilation efficiency of buildings
- transport of contaminants (between zones but also from outside)
- ventilation effectiveness

- pollutant removal efficiency
- age of air
- smoke propagation [1]
- assessment of ventilation heat losses
- passive cooling
- assessment of heat transport between zones

To know more about the different modules and their application refer to the COMIS fundamentals [2].

Many useful output options are available with COMIS 3.0: mass air flow rates per zone, mass flow matrices, outdoor air flow rates, air change rate, room mean age of air, building ages of air, air change efficiency building, room air change index, etc. They are extensively listed and defined in [3].

### EXAMPLE OF A CONCRETE APPLICATION

COMIS w. IISiBat has been used to model many buildings and their ventilation system. As a concrete example, figure 7 displays a Japanese house that has been modelled by Pr. Yasuo Utsumi.

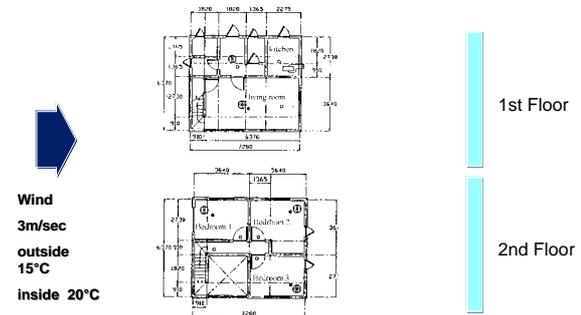


Fig. 7a: Example of a Japanese house modelled with COMIS w. IISiBat - floorplan

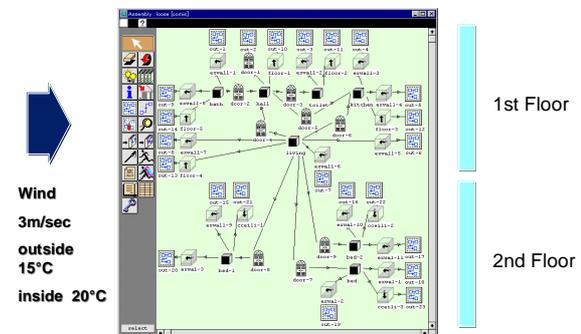


Fig. 7b: Example of a Japanese house modelled with COMIS w. IISiBat - representation in the IISiBat Assembly Window

This simple example of an application of COMIS w. IISiBat deals with the indoor air quality problem. Its goal is to check if natural ventilation is sufficient to assure sufficient indoor air quality for the given building. For this problem we are interested in *the air infiltration rate, the air ventilation rate and the air flow paths* inside the building. The *air exchange rate* can be derived from this output. The data needed by the system to calculate this information are :

- ELA<sup>4</sup> (building envelope including windows and interior doors)
- wind direction and wind speed
- plan area density, surrounding building height
- wind velocity profile exponent
- absolute humidity
- inside and outside air temperature
- barometric pressure

Further studies can then compare these results to a situation where mechanical ventilation is used to improve indoor air quality. This can be done by adding mechanical ventilation systems (HVAC components) to the IISiBat project description. For example, an exhaust fan could be added and connected between a room and an external node.

An example output for this problem is shown in figure 8.

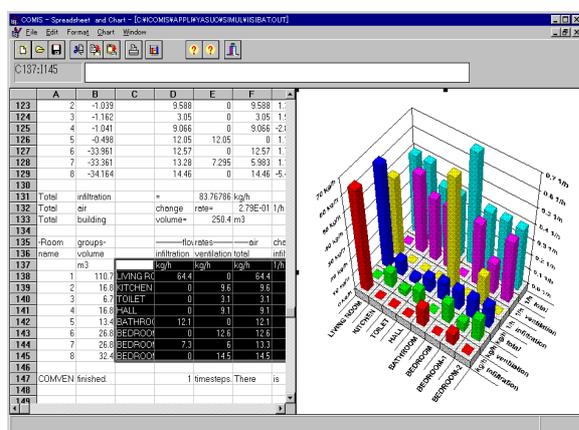


Fig. 8: Output

## SENSITIVITY ANALYSIS MODULE

In the frame of Annex 23-Subtask 3 (validation of COMIS), Jean-Marie Fürbringer at LESO worked on an advanced sensitivity analysis methods [6] to improve parametric studies and error propagation analysis as well. At the beginning of Annex 23, it was expected to improve the use of the COMIS

numerical models by incorporating a module for sensitivity analysis.

Different sensitivity analysis methods have been investigated. A simplified method (based on the Monte-Carlo approach) has been specified [8] and a prototype of this module has been developed and proved the interest of this utility. However, in COMIS 3.0, only a multi-run facility, leaving the analysis charge to the user, has been implemented. This multi-run tool facilitates parametric studies but does not solve the problem of error propagation analysis.

## COUPLING COMIS WITH THERMAL SIMULATION CODES

There are different coupling strategies:

- Sequential coupling is the most straightforward and also most time-efficient coupling method. It simply consists of invoking a first simulation tool, yielding a huge matrix where one dimension is time. This result is fed into another simulation run, invoking a second simulation tool. This coupling method, too crude for many problems, is sometimes sufficient.
- 'Ping-pong coupling' consists of alternately starting two simulation programs, each using data calculated by the other. Each program solves its own problem with its own solving method. The program that embeds the other acts as a supervisor to deal with global convergence problems. This is the method that has been used in Annex 23.
- Another method should yield the most accurate results; it consists of extracting the knowledge contained in various Building Performance Evaluators (as TRNSYS for example) and translating this physical knowledge into models suitable to a generic solver (domain independent). But this method induces such a huge amount of work that it is not very often used.

Within Annex 23, and in parallel to the development of COMIS 3.0, EMPA (Switzerland) derived a new TRNSYS Type (TYPE 57 see [4]) from COMVEN. This new type is designed to be connected to the TRNSYS multizone thermal model: TYPE 56. In a kind of ping-pong strategy, the calculation is done step by step using the results of one program as the boundary conditions for the other package.

<sup>4</sup> Effective Leakage Area

## RESULTS & DISCUSSION

To our knowledge, this is the first time that an IEA/ECB Annex produces software that can be widely used with such a confidence (thanks to the Annex 23-subtask 3 validation efforts). We hope that, through simulation environments as IISiBat, quite complex programs such as COMIS will be used by a larger number of people including engineers in consulting companies. And, if this is soon the case, we will have reached an important issue: make possible the transfer of simulation technology to practitioners.

## ACKNOWLEDGMENTS

As the Annex 23 subtask I leader, I would like to take this opportunity to thank all my partners in Annex 23 who brought enthusiasm and ideas, all those who developed modules, tested COMIS and made it progress. Special thanks to Hans Phaff (TNO -NL-), Viktor Dorer (EMPA -CH-) (who were, from far, the main COMIS module contributors) and Yasuo Utsumi (Miyagi College of Technology -J-) who helped us to simplify the use of the simulator.

There is not enough room to cite all the sponsors from the different countries who supported the research teams in their implication within Annex 23. Nevertheless, I would express my special thanks to the French Agency for Environment and Energy Management (Ademe) which agreed since the very beginning to support CSTB in leading the software development of COMIS 3.0.

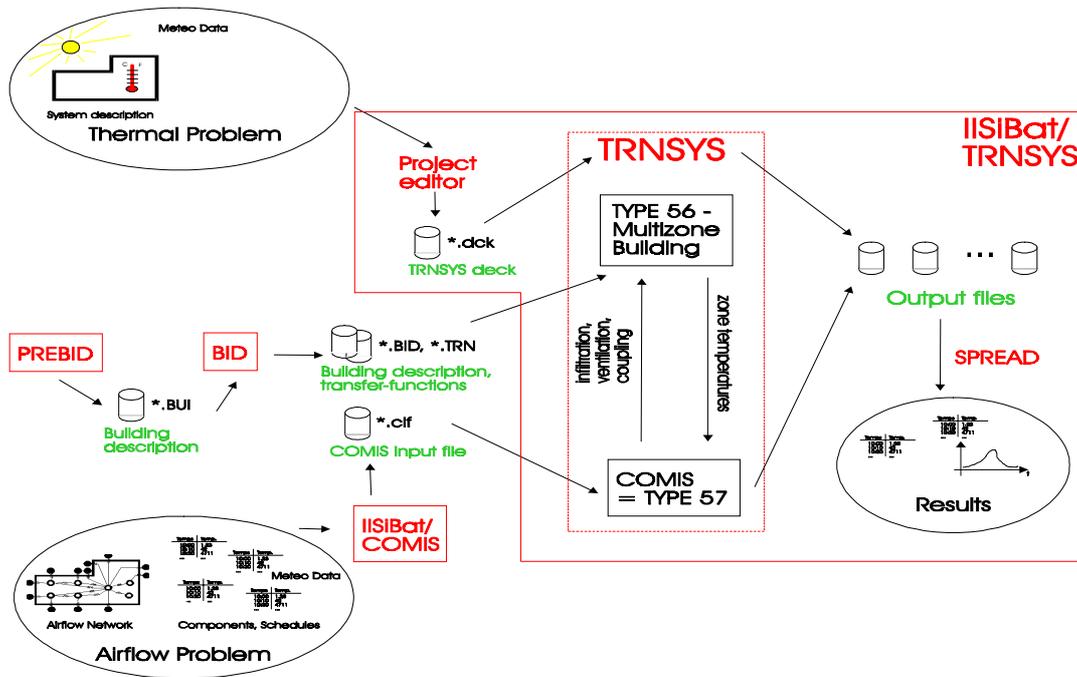


Fig. 9: Using COMIS with TRNSYS - the big picture

## REFERENCES

- [1] Borchiellini R., Cali M., Mutani G., 10/1993  
Evaluation to adapt COMIS to smoke movements in buildings analysis  
IEA.ECB.A23/93.10.13/RB
- [2] COMIS group  
Fundamentals of the Multizone Air Flow  
Model COMIS. 05/1990  
Technical Note AIVC 29.
- [3] Dorer V. & Weber A. 03/1995  
Output Options for COMIS.  
IEA/ECB Annex 23/95.03.15/VD
- [4] Dorer V. & Weber A. 06/1994  
Multizone Air Flow Model COMVEN-TRNSYS  
IEA/ECB Annex 23/94.10.13/VD
- [5] Dorer V., Huck F. & Weber A. 05/199  
PC based User Interface for Multizone Air Flow and Contaminant Transport Model COMIS  
IEA/ECB Annex 23/95.05.15/VD
- [6] Fürbringer J.-M. 1994  
Sensibilité de Modèles et de Mesures en Aéraulique du Bâtiment à l'Aide de Plans d'Expériences ;  
Ph-D document #1217, Ecole Polytechnique Fédérale  
de Lausanne (Switzerland). Dpt. de Physique.
- [7] Keilholz W. P. 12/1995  
IISiBat 2: an Open, Object Oriented Development and Simulation Environment  
Europ'IA 95 Conference. Lyon.
- [8] Keilholz W. P. 1996  
Développement d'un outil de calcul intégré multi-domaine, basé sur une approche orientée objets.  
Ph-D document CSTB/University of Nice
- [9] Pelletret R. Y., Soubra S., Keilholz W. 08/1995  
Transferring Simulation Techniques to End-Users. Application to TRNSYS.  
"Building Simulation '95", IBPSA Fourth Int. Conference. Madison.
- [10] : Sahlin P., Bring A. 1993  
Modelling Air Flows and Buildings with NMF and IDA, Proceedings of the IBPSA Building Simulation 93,  
Adelaide, Australia

References 1, 3, 4 and 5 can be obtained through the A23 Operating Agent: H. Feustel, Lawrence Berkeley Laboratory, Building 90, Room 3074, 1 Cyclotron Road, Berkeley, CA 94720, USA

Reference 2 can be obtained from the Air Infiltration and Ventilation Centre (AIVC), University of Warwick Science Park, Barclays Venture Centre, Sir William Lyons Road, Coventry CV4 7EZ, Great Britain