Intelligent Simulation Environments for Building Modelling and Simulation

S. Soubra and R. Pelletret

Centre Scientifique et Technique du Bâtiment
BP 141 - 06561 Valbonne Cedex FRANCE.
Tel. +(33) 93 65 34 00
FAX +(33) 93 65 29 37

Abstract
This paper describes the design of Intelligent Simulation Environments (ISE). An ISE aims to assist the user during the process of modelling and simulating of physical phenomena. Assisting the user means to automate repetitive tasks and, during creative and design tasks, to return a semantic feedback about the design. This semantic feedback will provide the user with the necessary knowledge and tools to develop a new model and/or to use an existing model. The general architecture is described and an example of an implementation is given.

1. INTRODUCTION

In the field of building design and construction, many modelling and simulation tools are available. These tools can cope with a large range of building related issues (e.g. energy consumption, air flow patterns and pollutant transport...) at different scale levels (e.g. global, detailed...).

These tools have been developed originally by researchers, and are generally powerful. Nevertheless, their spreading among user groups in building professions (e.g. designers, administrators, architects...) remains limited. The reasons for this are :

° these users must learn the input language of the tool they intend to use before being able to describe their simulation problem. This is generally felt as a difficult task, especially that most of the input languages are closer to programming languages than to the vocabulary that users in building professions are familiar with ;

° the input languages of the simulation tools are very different and these tools are unable to communicate with each other ; this means that the description of the same simulation problem for one tool cannot be used by another.

This also means that the description of different simulation problems (e.g. thermal comfort, air quality...) cannot share their common data even though they deal with the same "physical objects" (e.g. buildings, walls, openings, HVAC systems...) ;

° the users in building professions have different needs concerning simulation tools than researchers ; they expect to use "user friendly" tools that assist them during their problem solving tasks : they expect to be able to ask questions like : Where am I ? What did I do (wrong) ? What do I do next ? and have relevant answers to their questions.

An ISE aims to incorporate some form of contextual knowledge about the tasks it's performing and some form of knowledge about the user in order to offer guidance and counseling adapted to the user's needs and to facilitate the sharing of common data with other simulation tools.

2. GENERAL ARCHITECTURE OF AN ISE

To assure the functionalities of an ISE, several cooperating modules should be used (cf. Figure.1.).

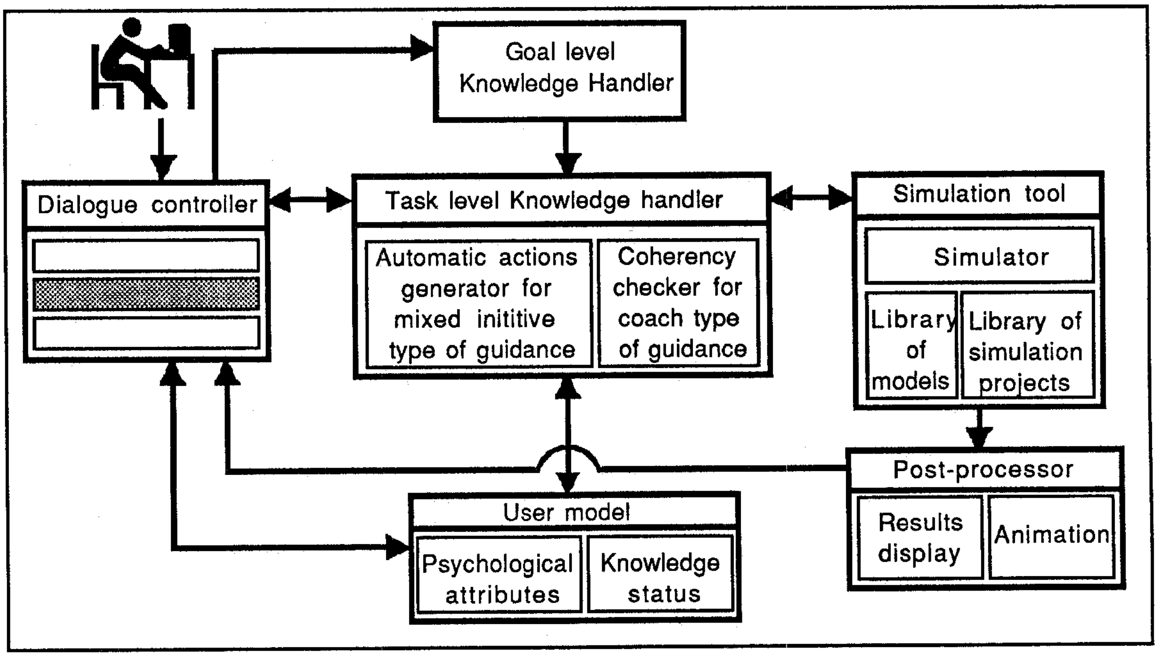

Figure.1. : Components of an ISE.

2.1. Dialogue Controller

User friendliness is (must be ?) a common issue to all interactive applications : "Poorly designed interfaces can include degraded user productivity, user frustration, increased training costs and the need to redesign and reimplement" [1]. Without going through the details of Human/Computer Interactions, we will state some of its aspects.

Like all human behavior, computer interactions involve three types of basic human processes : perception, cognition and motor activity. The interface designer's job is to design interaction techniques which minimize the work required by these processes both individually and in combination.

Concerning the perceptual process, the techniques that an interface designer can use include color coding, blinking, movement, reverse video... to call attention to specific parts of the display. For the motor process, issues include the choice of a pointing device (stylus, tablet, mouse...).

The cognitive process is more complex than the two others and therefore more difficult to organize. One rule that seems to emerge (detailed in [2]), is that when we learn how to use an interaction technique, we acquire and organize information concerning its use. If this information fits into categories of concepts we understand, then learning can proceed rapidly ; if not, learning is slower and recalling what we learned is harder. Interface designers should try to make a link between new concepts used by the interface and concepts users already know. Metaphors can be a mean to do this link (for example the "Desktop metaphor" used in the Macintosh interface).

The Dialogue Controller is the ISE component which takes in charge the conversation with the user, in a manner tailored to his conceptual understanding of the observed phenomena. In addition, the Dialogue Controller must be able to handle a wide range of interactive styles. The choice between these styles will be dynamically made in accordance to user's preferences as determined by the user model (cf. §2.2). For example, the ISE must be able to handle different model description formalisms (block-diagrams, bond graphs, electronic schematic diagrams...). This feature is called multiformalism. It means that several formalisms can be used separately but also in conjunction with each other.

2.2. User Model

The user model is the ISE module which contains an explicit representation of the user's psychological attributes and knowledge status. Psychological attributes are a set of informations about the user preferences and cognitive capabilities. By knowledge status we mean both domain related knowledge (knowledge about building modelling and simulation) and tool related knowledge (knowledge about the functionalities of the simulation tool).
The psychological attributes are long term user characteristics. They might be considered static. On the other hand, knowledge status is a variable characteristic of the user. It will change as the user progresses and learns new techniques and sophisticated strategies for problem solving.
The knowledge status component of the user model could be dynamically updated by the system during the dialogue and according to the user's behavior.

2.2.1. Links with other ISE components

The user model would supply other ISE components with the following information :

° the Dialogue Controller (cf. §2.1) will use the psychological part of the user model to determine what "style" of user interface is best suited to the user preferences (a style of a user interface is, for example, the model description formalism used) ;

° the Task Level Knowledge Handler (cf. §2.3.2) will use the domain related knowledge status to personalize "on line" help messages and warnings. This means that when the user does an action that is considered by the system not coherent with the initial posted goal (cf. §2.3.1), the system will explain "why" the action is not coherent using a set of concepts familiar to the user.

2.2.2. Behaviorist User Model

User modelling is a complex field of research at the leading edge of AI and cognitive science. The specifications for a user model that can reproduce the complex human understanding and reasoning processes do not exist at the moment. It was thought that a user model could be a network of concepts representing the expert's knowledge of the domain, and that, as novices accumulate experience, their knowledge moves along this network until they reach the expert's knowledge level. This approach was proven to be wrong. In fact, novices and experts conceptualize a domain in radically different ways, and learning is not a process of accretion of knowledge towards the expert's view but involves progressive reconceptualizations [3]. This suggests a need for a sequence of domain representations ranging from a simple qualitative view to a detailed quantitative view. The user model described below takes these considerations into account.

The user model could be, as a first step, a model which estimates the knowledge level of the user concerning the tool. In this case, the inputs of the model $\{i_j\}$ would be a set of user's actions during a working session and the outputs $\{o_j\}$ would be inferred dynamically and represent the changes in the user interface of the simulation tool.

A first set of user's actions taken into account $\{i_j\}$ could be the following :

° the duration of use of the tool ;
° the frequency of use of some of the functionalities of the tool ;
° the frequency of manipulating errors.

Taking into account these parameters, the user model would communicate to the dialogue handler the needed information to adapt the functionalities of the tool to the new level of the user's knowledge. These changes would reflect the improvements in the user's conceptualization of his work.

A first set of possible changes in the user interface {oj} as the user acquires knowledge about the simulation tool, could be the following :

° the representation on the screen of the physical components could become more abstract ;
° the functionalities of the ISE could become powerful (thus more complex) ;
° the coherency check of the user's actions could take place less often and the number of warnings could be reduced.

One possible method to establish relations between the {ij} and the {oj} is to use empiric relations based on the observation of the use of simulation tools. These relations could be validated in a later phase. They can also be modified by the system in accordance with the user's reaction about the changes. Since the {oj} are inferred dynamically, a coherency check over the whole set seems necessary so that incompatible changes in different aspects of the user interface do not occur.

To this behaviorist user model, we could add a psychological component allowing the user to specify his preferences. This component could be explicit and static. Thus its contents could be specified directly by the user and would not change with time.

The development of a simple behaviorist user model could be done at relatively low costs when the {ij} and the {oj} are a small subset of all the factors in play.

2.3. Contextual Knowledge Handlers

The separation between the task level and the goal level means asking users to communicate their goals in addition to the different tasks during their problem solving process. This is important for two reasons :

° it encourages the users to explicitly formulate their goals. This leads them to a better focus on the strategic level of the physical phenomena studied. This corresponds with Singley's findings [4] : "students performance on solving certain calculus problems improved if they tried to post the goals they were working on before selecting any specific operators" ;

° it provides the system with crucial information for user modelling which is often difficult to induce from the operations themselves.

2.3.1. Goal Level Knowledge Handler

One possible method for determining the final goal of the modelling and simulation process is to let the user describe it explicitly. This leads to a complex interpretation of natural language.
Another method is to deduce the final goal from the user's intermediate actions. This method is complex and error-prone ; especially at the early stages of the problem solving process because, at this stage, only little information is available. This is an important drawback of the method because

the user usually needs guidance at the early stages of his work where the most important choices are made. But, at this stage, and according to this method, since the system doesn't have any knowledge about the user's goal, it cannot offer him guidance.

Instead we propose a technique which uses a small subset of natural language to describe the user's goal. This provides useful and easily interpretable information.
This technique is based on the hypothesis that the expression of a physical problem using terms related to a modelling and simulation approach can be decomposed into three criteria :

- the physical objects in play (e.g. buildings, HVAC systems...) ;
- a time criterion which is a notion about both the duration and the frequency of the observed phenomenon ;
- the simulation criterion (e.g. energy consumption, comfort conditions...).

The final simulation goal can therefore be described by the user on the basis of a composition of those criteria using menu selection techniques. This goal is then transmitted to the Task Level Knowledge Handler.

2.3.2. Task level Knowledge Handler

The information coming from the Goal Level Knowledge Handler will be used by the Task Level Knowledge Handler in two ways :

- it will determine what are the sets of models necessary for reaching the posted goal (different sets of models might be used to reach the same final goal). The determination of these sets of models would be based on the necessary output needed. For example if the simulation criterion is the comfort conditions in a room, the used set of models must have the inside room temperature as one of its outputs ;
- it will deduce what are the models that should not be used or whose use is not relevant. The determination of these constraints would be based on two aspects :
  - the relation between the time criterion as specified in the posted goal and the level of detail of the model. For example, if the simulation criterion is an estimation of annual energy consumption, it wouldn't be justified to use a detailed model ;
  - the relation between the posted goal and the underlying physical hypothesis of the model. For example if the simulation criterion is the study of comfort conditions, it wouldn't be relevant to use models with a steady state hypothesis.

To make these tests, an ISE needs information about the available models. This information could come from PROFORMA data files [5]. A research work in which the CSTB is taking part is going on about the models documentation in view of model reuse and model sharing ; the "standard" documentation of a model in this work is called a PROFORMA.

The other functions of the Task level Knowledge Handler which are not related to the posted simulation goal but which use the PROFORMA data files to make a coherency check of the assembling concern :

° the connections between input and output variables as compared to their units ;

° the values of the parameters as compared to validity ranges determined by the author of the model.

The task level could have two types of guidances strategies among which the user could choose the strategy that suits him best :

° coach strategy of guidance : in this case, if the user's actions are assumed to be coherent, the system remains transparent. Otherwise, the user is warned that the choice he just made is not coherent with the initial formulation of the simulation goal. At this stage, the user can either update the formulation of his goal or undo his last action.

° mixed initiative type of guidance : in this case, the user could ask the system for more guidance. The system could then propose an assembling of models corresponding to the posted goal and the constraints of coherency with the already existing models.

2.4. Simulation Tool and Post-processor

The simulation tool (simulator) is the component concerned with the calculation phase. For building modelling, the simulators generally deal with what is called analog simulation (where descriptive variables can be calculated at any time instant) which is to be distinguished from discrete simulation (where descriptive variables are available at discrete-valued time instances only).

Since the choice of a simulator is often a compromise between two aspects :

° the functionalities of the simulator as compared to the characteristics of the problem under study (discontinuities, type of the equations, stability...) ;
° the availability, support and maintenance of the product ;

we consider that an ISE should be simulator independent i.e. the user should be able to choose between the available simulators the one that suits his problem best.

In any case, the simulator should contain libraries of models and libraries of simulation projects. The libraries of models could be used to do an assembling of models representing a complex system (the models in this case would be considered as the elementary bricks used in a "modelling by assembling" approach). The simulation projects are predefined assemblings of models for future simulation sessions.

The users of these two types of libraries are not necessarily the same. The users of the libraries of models would be concerned in the modelling process. On the other hand, the users of the libraries of simulation projects would be essentially concerned in doing parametric studies and validations. This shows that an ISE should be adaptable to the working methods of a team on a first level and, on a second level, should be able to influence their working methods in order to transform the models in "reusable components". Influencing the working methods in the modelling process can be done by defining guidelines for this process. The standard documentation of the models is one of these guidelines, the distinction between different types of libraries is another one.

The post-processor is the component that displays the simulation results. It could be a separate component or a simulator related component. The functionalities of the post-processor could be :

° to give the user a large range of choices for displaying the simulation results (e.g. histograms, plotters...) ;

° to show a dynamic visual representation (DVR) of the state of each model during the simulation (the state of a model could represent for example the value of one or more of its variables). What we call a DVR is not an animation during the simulation but is a symbolic representation of the state of each model which moves between two extreme values. This representation could be done with a gauge symbol whose indication would change dynamically during the simulation. This issue seems important not only for display reasons but also for validation reasons. For instance it would be easy for the user of an ISE to see that a model is not changing its state as it should be (going higher or lower than the extreme values...) because it's simply visible on the screen. This shows not only what happens in a system but how it happens. A DVR approach seems quicker and safer than the usual method of analyzing the results of the simulation.

3. IMPLEMENTATION ISSUES

A first application of the general concepts of ISE has been developed in the frame of the IISIBât project. IISIBât is implemented on a workstation and the programming environment is the graphical toolbox Aïda and MIPS (an Object Oriented layer over Le_Lisp developed by the CSTB) [6]. IISIBât uses TRNSYS as a simulation tool [7]. TRNSYS (a Transient System Simulation program) is a simulator designed by the Solar Energy Laboratory of the University of Wisconsin to assess building thermal performances. Another application for multizone air flow modelling and pollutant transport is being realized in the frame of Annex 23 of the International Energy Agency with COMIS as a simulator (COMIS - Conjunction Of Multizone Infiltration Specialists - is a simulator developed by a multi-national team to deal with multizone infiltrations) [8]. In this frame, the sharing of common data between the models of TRNSYS and the models of COMIS will be tested.

The Knowledge Handler implemented in IISIBât is a Task level Knowledge Handler. A Goal level Knowledge Handler is under development on the basis of the simulation goal acquisition technique presented (cf. § 2.3). The Task level Knowledge Hander uses the documentation related to the models as a source of information for the checking functions. This documentation is based on the PROFORMA (cf. § 2.3.2). Information stored in a PROFORMA file consists on a description of the inputs, outputs and parameters of the model (names, definitions, validity ranges, default values...). An illustration of the coherency check over the connections between input and output variables as compared to the semantic attached to each variable is given in Figure 2. A checking function based on the semantic attached to variables concerns the units but also the definitions of these variables (for example the connection between a Air_Temperature and a Surface_Temperature is supposed to be wrong even though these two variables have the same unit).

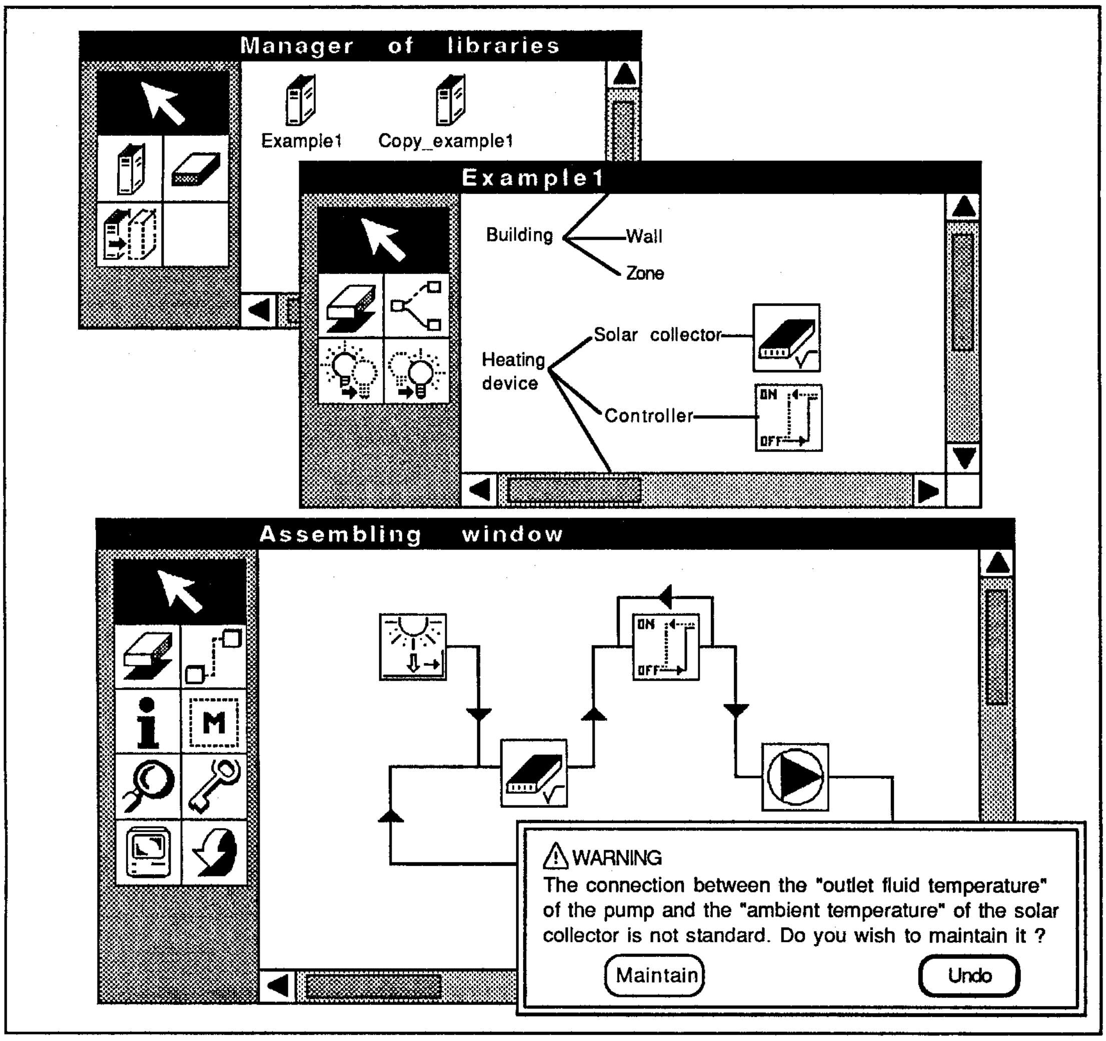

Figure.2. : A working session with IISIBât.

CONCLUSION

An ISE is a user-friendly simulation tool capable of offering to the user relevant guidance during the modelling and simulation process. This means that an ISE must have, on one hand, some knowledge about its user and, on another hand, some knowledge about the modelling and simulation process.

In this paper we presented the general architecture of an ISE and the functionalities of its components. In particular, a Behaviorist User Model has been described. The importance of the distinction between the task level and the goal level knowledge handlers has been underlined. A proposal for a simulation goal acquisition technique has been made. An example of an implementation has been given.

REFERENCES

[1] FOLEY J.D., WALLACE V.L. and CHAN P. *The human factors of computer graphics interaction techniques*. IEEE CG&A. November 1984. p. 13-47.

[2] SOUBRA S. and PELLETRET R. *The IISIBât project*. Séminaire AFME "Outils d'aide à la conception et la gestion". October 26-27, 1989. Sophia-Antipolis. France.

[3] SELF J.A. *The use of belief systems for student modelling*. Artificial Intelligence and human learning. 1988. London. Chapman and Hall.

[4] SINGLEY M.K. *The effect of goal posting on operator selection*. Proc. 3rd Int. Conf. on AI and Education. 1987. Pittsburg.

[5] DUBOIS A.M. *Le PROFORMA*. CSTB internal report n°MGL/787. 1987.

[6] POYET P. *Integrated Access to Information Systems*. Applied Artificial Intelligence 4, 3. 1990. p. 179-238.

[7] User guide. *TRNSYS. A transient simulation program*. University of Wisconsin. Madison. 1979.

[8] User guide. *COMIS*. Edited by the Lawrence Berkeley Laboratory. Berkeley. 1991.

[9] DAVID E. *The role of Mental Models in learning to operate a device*. Cognitive Science 8. 1984. p. 255-273.

[10] HOLLAN, J.D., HUTCHINS, E.L and WEITZMAN, L. *STEAMER : an inspectable simulation based training system*. AI magazine, 5, 2, 1984. p. 15-27.