Building Simulation

Seventh International IBPSA Conference
Rio de Janeiro, Brazil
August 13-15, 2001

A SIMPLE INTERFACE TO CFD CODES FOR BUILDING ENVIRONMENT SIMULATIONS

Charles R. Broderick, III and Qingyan Chen
Building Technology Program
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

ABSTRACT

It is becoming a popular practice for architects and HVAC engineers to simulate airflows in and around buildings by Computational Fluid Dynamics (CFD) methods in order to predict indoor and outdoor environments
. However, many CFD programs are crippled by a historically poor and inefficient user interface system, particularly for users with little training in numerical simulation. This investigation endeavors to create a Simplified CFD Interface (SCI), a public domain program that allows architects and building engineers to use CFD without excessive training. SCI can be easily integrated into new CFD programs.

INTRODUCTION

Advanced building design requests information about airflow in and around buildings. The information concerning airflow in buildings are air velocity, temperature, relative humidity, and contaminant concentrations that are important to assess thermal comfort and indoor air quality. The information on outdoor airflow are mainly air velocity and pressure distributions that are crucial for thermal comfort and building structure designs. Traditionally, the information is obtained by experimental measurements in an environmental chamber for indoor airflow and in a wind tunnel for outdoor airflow. The experimental studies are expensive and time consuming. On the other hand, developments in computer technology and turbulence modeling enable designers to obtain the information by Computational Fluid Dynamics (CFD).

The CFD technique requires turning a building physical model into a numerical model with which a computer can solve to generate the information needed for indoor and outdoor environment design. Historically, the conversion was normally done by highly skilled scientists and engineers who know the detailed governing equations used in CFD and the corresponding numerical techniques. The CFD results were mainly in ASCII format, although some dedicated computer packages could turn them into tables and graphical charts. In recent years, the development in computer technology, especially graphical software brings convenience to the CFD users. The CFD users are no longer CFD experts. Architects and building designers have attempted design indoor and outdoor environment with a CFD software interface that can convert a building physical model into a numerical model and can present their CFD results as graphics.

Nevertheless, many CFD software interfaces were originally designed for users with a strong fluid dynamics background to maximize the CFD capacity, allowing the users to directly alter multiple esoteric numerical simulation parameters. As simulation software grew more and more advanced, the variety and exoticness of parameters available for tuning increased in number. The user interfaces, in turn, responded to the additional complexity by adding additional fields into already-crowded user interface dialogs. While unpleasant, this solution was acceptable; at the time, users were typically experts in numerical simulation, well-versed in its methodology and not prone to confusion. However, as the user base shifted from CFD experts to architects, building engineers, and other novice CFD users, the maximal-utility design often ensnares the user in confusion and encourages poor decision-making.

The rich complexity and multitude of parameters is only one obstacle facing novice CFD users. Occasionally, simulation software will fragment the three tasks required to complete a simulation (model creation, CFD solution, and results visualization) into three separate programs, which are in turn controlled by a separate main launching program. Even worse, some simulation software offers only two of the three necessary programs, relying on the user's experience with third-party software to finish the task. Learning and remembering three separate user interfaces distracts the user's attention from what is most important: visualizing accurate results quickly.

While novice CFD users are familiar with computer assisted tools through other work, unassisted CFD simulation represents a level of computational complexity significantly beyond their existing skill set and cannot be quickly acquired. Examples of some computer-assisted tools comfortably used by our user base include illumination simulators, CAD programs, and acoustical simulation software. Some of the most successful of these tools are plug-in modules to

AutoCAD, a popular full-featured drafting program. These non-CFD modules are successful because their data input models and simulation results can be easily edited and viewed using the familiar AutoCAD user design environment. Unfortunately, airflow analysis and other CFD solvable problems, such as pollutant dispersion simulation and thermal comfort prediction, require specific and more detailed model characterization than an AutoCAD plug-in could comfortably provide without overextending its intended reach.

SCI is a public domain program designed to alleviate the stresses placed on novice CFD users. This goal is achieved through following three system design methodologies. First, the number of esoteric parameters available to the user is restricted through forced default values or elimination. Second, the steps required to create, run, and view a simulation are combined into a single, small, Microsoft Windows-compatible application, using user interface paradigms that all PC users are familiar with. Third, the type of input accepted by our interface is broadened to include formats the new user base would find most useful, such as industry standard file formats and CAD files. Finally, back-end generality is provided through support for multiple CFD engines, eliminating the need for time-consuming model regeneration. Combined, these methodologies create user interface benefits similar to AutoCAD plug-ins while still supporting complex CFD analysis. Users may become accustomed to SCI's simple, single front user interface, while concurrently accessing a powerful range of CFD engines at the back end.

RESEARCH APPROACH

Intelligent Parameter Reduction

CFD software's rich parameterization can be categorized into two distinct sets. One set of CFD parameters, called the *physical parameter set* as shown in Figure 1(a), contains real-life attributes of the building layout. Examples include characterizations of objects in the building (such as building geometry, a television set with a certain temperature, or the window with a certain heat influx) and flow conditions in the buildings (such as the airflow from an HVAC system). The second set of CFD parameters is the *computational parameter set* as shown in Figure 1(b). These variables include the computational mesh topology, error thresholds, turbulence models, phantom computational steps, and solver relaxation factors. These parameters help the underlining CFD software converge quickly to an accurate solution.

Architects and building engineers deal with components of the physical parameter set on a daily basis. These parameters are well understood, and users have little problem providing useful values or understanding their effect on the results. Reducing or removing these parameters from the interface does not result in a significant decrease in user confusion. However, understanding the purpose of the computational parameters requires knowledge beyond that of the user base. Intelligent groupings, default values, and parameter elimination are used to reduce and combine many of the parameters that obfuscate this part of CFD design work. The CFD interface acts as a *knowledge bridge* as shown in Figure 2: on one side are the terms and figures familiar to architects and building engineers, and on the other is the knowledge required for a successful CFD simulation. Previously, a CFD specialist interacted with both sides to ferry the model definition and simulation results back and forth. Now, this process is automated by incorporating the CFD specialist's intelligence directly into the interface.

For example, some definition tools contain labyrinthine user interfaces for creating mesh topologies as shown in Figure 3(a), due to the number of mesh variations and possible geometric subtleties. However, novice CFD users may be concerned more with viewing useful results within a rapid time frame than about topology theory. Therefore, they could be willing to sacrifice small (but measurable) amounts of accuracy for faster, simpler mesh definition tools. SCI's definition tool is designed solely for rectilinear, variable spaced meshing as shown in Figure 3(b). In general, such a mesh simplification would not lead to an error great than 15% than that with body-fitted coordinate systems. This kind of mesh is easily and clearly specified from a minimal amount of user input and remains powerful enough to achieve dense, accurate results in the model's important regions.

Further parameter reduction can be achieved by restricting variations in fluid conditions. While the underlining CFD software might be capable of analyzing flow patterns in air as well as in water, novice CFD users in building environment model may be not likely to find the water option useful. To successfully estimate default values for ambient air, the altitude of the building site is requested. Building site altitude is a value well known by our users and is easily transformed into the gravity, pressure, and viscosity terms CFD simulation software desires. This kind of parameter reduction is a highly successful example of a knowledge bridge. The interface provides the intelligence necessary to seamlessly transform the user's intuitive understanding of the model (the altitude of the building site) into values

useful for CFD operation (a complex characterization of the ambient air).

A Unified Platform for CFD Simulation

Older CFD simulation programs fragment their user environment into three separate programs: model generation, model simulation, and model visualization. These subprograms then share data through a common file space or through information pipes. This program flow provides the most benefit to users who have fleshed out their numerical design in advance, and penalizes users who switch back and forth between subprograms. However, most architects and HVAC engineers desire interactivity, unity and fluidity between the model generation, CFD computation, and results visualization stages. In particular, the following five stages of CFD simulation design as shown in Figure 4 are commonly executed.

In the *initial generation* phase, the user gives a shell description of the problem to be simulated. This typically includes specifying a small subset of the physical and computational parameter sets, including adding the room's objects and indicating the level of computational accuracy desired. The *computation* phase is handled by the CFD engine chosen by the user. The results are then forwarded to the *visualization* phase, where the user analyzes the results. At this point, the user may choose to enter the *model modification* phase and solve again, or save the results for the *presentation* phase. Examples of slight model modifications include minor alterations to the room geometry, changing airflow rate or direction, increasing the density of the mesh topology (to investigate a particular flow detail), or altering other parameters to achieve finer results. This investigation endeavored to create an interface that allowed rapid and seamless transition between the five stages of CFD simulation design.

SCI combines all three applications into a single workspace, eliminating any wasted time or effort incurred by application switching. To achieve this result, a window with a large, central, blank area is created for 3-D visualizations and presentation animations. There are two dialog boxes in the interface. The first dialog box controls the visualizations viewed in the main window, and the second dialog box commands the model layout tools. A generic menu bar controls all other parameterizations with four separate tools: a mesh topology tool, a problem description tool, an iteration control tool, and a simulation properties tool. Activating any of the tools launches dialog boxes that assist the user through the process. To avoid workspace clutter, these boxes disappear as soon as they are no longer needed. Moving to the computation stage requires the pressing of a button marked "GO" on the toolbar. Thus, all five stages are represented on the main window, less than a single mouse-click away.

Automating Model Format Translation for Rapid Model Development and Multiple Simulation

Many professional CFD packages require their users to specify the layout of the room using a foreign or unfamiliar methodology. Sometimes the topology of the mesh is used as a physical reference to locate physical items, such as walls, tables, and chairs (this is known as a *logical coordinate system* as indicated in Figure 5(a)). At other times, the user is prompted for the precise distance of objects from a common reference point (known as a *physical coordinate system* as illustrated in Figure 5(b)). If a user switches from a CFD solver that uses the former system to one that uses the latter, then the user must recast the model's object into the new format. In many cases, whenever the user switches CFD simulation software, he or she must reconstruct the entire building design using the methodology of the new CFD software.

This process of reconstruction impedes the success of many CFD user interfaces. Two immediate solutions to this problem involve automating the process of transforming the model from one CFD software data format to another. This first solution is to adapt a building environment file format standard. An international consortium of programmers known as the International Alliance for Interoperability has established a standard building file format known as IFC, or Industrial Foundation Class, to realize this solution. Building environment simulators, such as CFD simulators, are to include the ability to automatically translate IFC files into their own desired format, thus eliminating the necessity of involving the user beyond the initial generation. The other solution is to only ever use one single interface, and hope the interface can internally handle translations needed to run the different CFD solvers. This solution is similar in essence to how AutoCAD plug-ins behave – AutoCAD is the single interface where a model is created, and the plug-ins give and receive model data to the interface. The first solution is a *smart-application* solution and the second is a *smart-interface* solution, since the first requires additional intelligence built into the simulation software and the second requires additional intelligence built into an umbrella interface, as shown in Figure 6.

Figure 6 also shows that SCI employs both solutions. A translator for reading IFC-formatted files and converting them to our own, internal SCI format is included to implement the *smart-application* solution. In addition, an IFC exporting utility is included,

should the user want any geometry changes made in SCI to be recognized by other IFC-compliant building simulation tools. In addition to IFC compliance, we also included a STL (Stereolithography) geometry file [1] translation tool, to read model geometry directly from AutoCAD.

The implementation of the *smart-interface* solution allows SCI to work with more than just one underlying CFD numerical simulator. Once the model is translated into the SCI format, we can send it to as many CFD simulators as we have written translator modules. Currently, we have two translator modules. The modules are an interface for a CFD code to provide SCI data in a specific format. The modules can be written in any program language.

RESULTS

The mesh generation tool generates a uniform, x × y × z mesh in less than ten mouse clicks. Non-uniform meshes are created through additional mesh *regions*. A mesh region is an area of uniform or exponential nodal spacing. By linking different regions of different uniform or exponential spacing along the same axis, a non-uniform mesh topology can be created. Thus, this tool exhibits a useful duality – simple meshes can be constructed very rapidly without sacrificing the ability to build more powerful meshes at a later time.

The model layout dialog box in the top right hand corner of the window allows the user to insert objects such as windows or desks and lamps into the simulation. These objects are specified using the physical coordinate system, since this is the universe typically dealt with by the user base. However, to accommodate the most useful logical coordinates, the user has the option of specifying the edges of the computational mesh as the start or end locations for objects. This is very useful when inserting a window into a building model, since these commonly need to lie on the edge of the computational mesh. Should the user need to modify the physical size of the computational mesh, there is no need to go back to the model layout – the building's window will move with the mesh.

Typically, the user has a long list of objects to insert into the simulation model. For example, a floor plan might have three rooms, each with two lamps, a table, and a computer. Instead of listing all these objects in one long list, the user is given the option of creating and managing folders of objects. For example, a SCI model could have three folders, one for each room. Inside each folder lie the objects for that room. As the user clicks on different folders, all the objects contained underneath the folder are highlighted green on the main window, to remind the user which objects are contained in which folder.

The model layout dialog box also has the notion of *activating* or *deactivating* an object or a folder. An object can be activated or deactivated by selecting it and clicking on the activation button located on the top of the dialog box, as shown in Figure 7. A deactivated object will not be included in the model simulation, but can be reactivated at any time without reentering the location and parameter information again. Deactivating a folder deactivates all objects and folders contained within it. Following the floor plan example, the user might have two folders – floor plan one and floor plan two – and within each of those folders could exist a different floor plan. The user could evaluate one floor plan by deactivating the second, and vice-versa, quickly and effortlessly.

In the problem specification tool, the user indicates the turbulence model and the results the CFD simulator should solve for, as shown in Figure 8. The property specification tool asks only for the altitude of the building site. The final parameter window sets the iteration control. Since iteration control deals solely with computational parameters, it is the most confusing of all the tools. SCI eliminates the necessary parameters down to three values per variable, which are crucial to assisting the underlining CFD simulator in reaching convergence. For the users benefit, realistic default values are specified.

After defining the computational mesh and a few objects, the user is ready to compute airflow. Solving a room with two windows and a heated object in the middle of the floor requires less than 35 mouse clicks.

When the CFD simulator is finished, the user can initiate visualizations through the dialog box on the bottom left. The visualization dialog box allows only enough plotting options to support what most architects and designers need for their particular brand of data. We allow five different kinds of plots: mesh (to view the computational mesh created by the generation tools), pseudocolor (also known as false color), vector, boundary (to view the outlines of objects in the model), and contour. Multiple plots can be layered on top of each other to provide a large variety of presentation styles and effects. Each plot also has its own parameterizations, such as color, line thickness, data source, and, for 2D plots, slice location. In addition, some plots are equipped with an animation button, to allow the architect to realize a full volume impression of the flow within the model. Figure 9 shows a typical visualization window.

The user always feels as if he or she has control over the editing and viewing of the model, since both are

packaged tightly together. He or she only feels the numerical simulator with some distance; a side effect from the back-end generality. Since SCI has no control over the numerical simulator, the look-and-feel of this stage can vary significantly.

PROGRAM INTERNALS

SCI is written in Visual C++ using MFC technology for the user interface [2]. As such, it is an object-orientated application, specifying and implementing roughly 80 objects. At the highest system-design level, SCI resembles the knowledge bridge discussed above. It interacts with the user and storage repositories on the architecture user's side, and with a wide range of simulators on the CFD side.

The IFC database object is composed of the BS-Pro IFC client [3]. This module is sponsored by the International Alliance for Interoperability [4] and implements the interface for reading and writing IFC files.

SCI maintains its connection to the user through graphics calls to OpenGL [5] and through MFC objects. The IFC and STL translation modules, as well as the simulation format translation modules that communicate with the various different CFD solvers, transit information through the internal SCI database, as shown in Figure 10. The database is implemented using a combination of the standard template library and custom made objects. The simulation format translation modules are where most of the intelligence for module mapping is kept, and the MFC objects maintain the simplicity and unity of the user interface.

DISCUSSION

SCI has been used as a common CFD interface for three different CFD engines: a simple CFD program with a zero-equation model [6], a general CFD program with various viscous turbulence model, and a large eddy simulation program. The CFD programs were developed by different people at different times. They can be easily linked to SCI to do different simulations of indoor and outdoor environments. SCI eliminates the burden of creating a new user interface for every CFD program from scratch.

SCI has been used to train hundred architectural and engineering students at the Massachusetts Institute of Technology. Although the backgrounds of the students are very different, the students can use SCI to do simple indoor and outdoor environmental analysis with a two-hour training.

CONCLUSIONS

The research has developed a Simple CFD Interface (SCI) for architects and building engineers to obtain flow information in and around buildings for thermal comfort and air quality analysis. SCI uses a language familiar to the architects and building engineers by eliminating highly technical terminologies used by CFD experts. SCI can easily convert a building physical model into a numerical model for CFD simulation, activate different CFD engines, and informatively present the CFD results.

ACKNOWLEDGEMENTS

This study is supported by the National Science Foundations through grant CMS-9623864.

REFERENCES

[1] StereoLithography interface specification. http://www.Ennex.com/fabbers/StL.sht, Ennex Corporation, 1989.

[2] Visual C++/MFC frequently asked questions. http://msdn.microsoft.com/library/backgrnd/html/msdn_mfcfaq50.htm, Microsoft Corporation, 1997.

[3] Laine T, Kosonen R, Hagström K, Mustakallio P, Yin DW, Haves P, Chen Q. 2000. Better IAQ through integrating design tools for the HVAC industry. Proceedings of Healthy Buildings 2000, Vol. 4, pp. 279-284, Espoo, Finland, 2000.

[4] IFC specifications development guide. International Alliance for Interoperability, 1997.

[5] OpenGL specification. http://trant.sgi.com/opengl/docs/Specs/glspec1.1/glspec.html Version 1.1., Silicon Graphics Inc., 1997

[6] Srebric, J., Chen, Q., and Glicksman, L.R. 1999. "Validation of a zero-equation turbulence model for complex indoor airflows," ASHRAE Transactions, 105(2), 414-427.

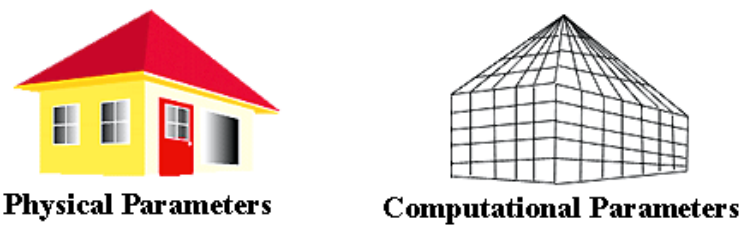

Figure 1: The physical parameter set includes information about the physical geometry of the problem. The computational parameters concern the numerical simulation of the model.

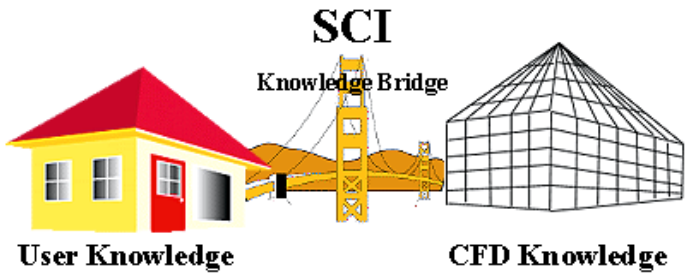

Figure 2: SCI acts as a knowledge bridge between our users and CFD simulation.

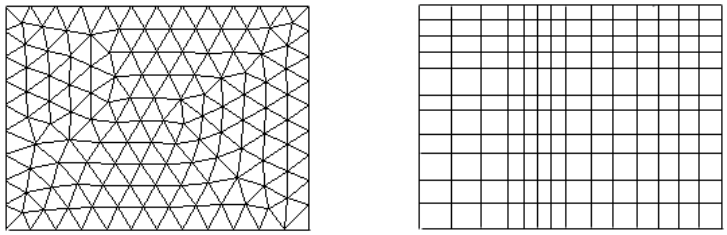

Figure 3: Mesh topology can become very hard to specify. Constant spaced rectilinear meshes, on the other hand, remain simple and powerful.

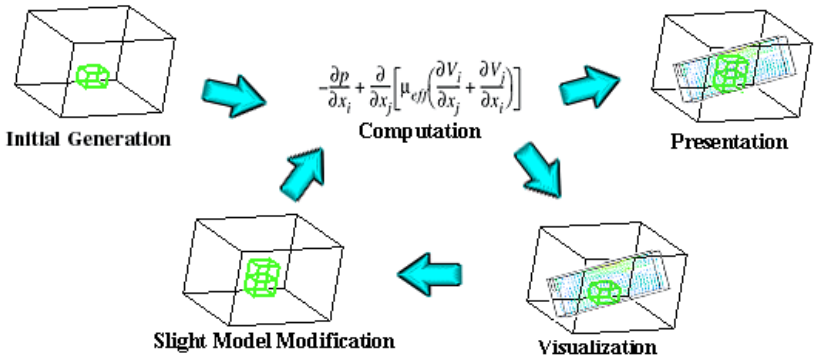

Figure 4: A common program flow loop for CFD design work.

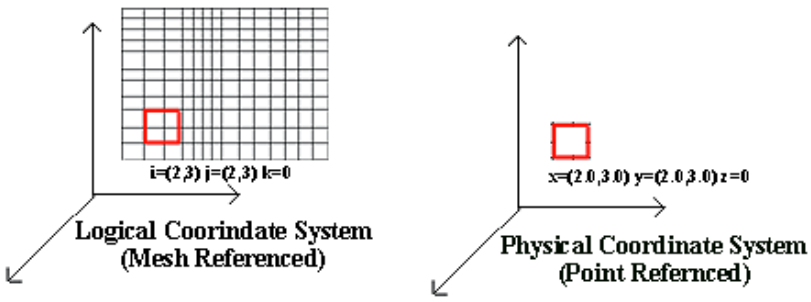

Figure 5: Logical coordinate systems use the computational mesh to locate objects. Physical coordinates use offsets from a reference point (usually the origin of the coordinate system).

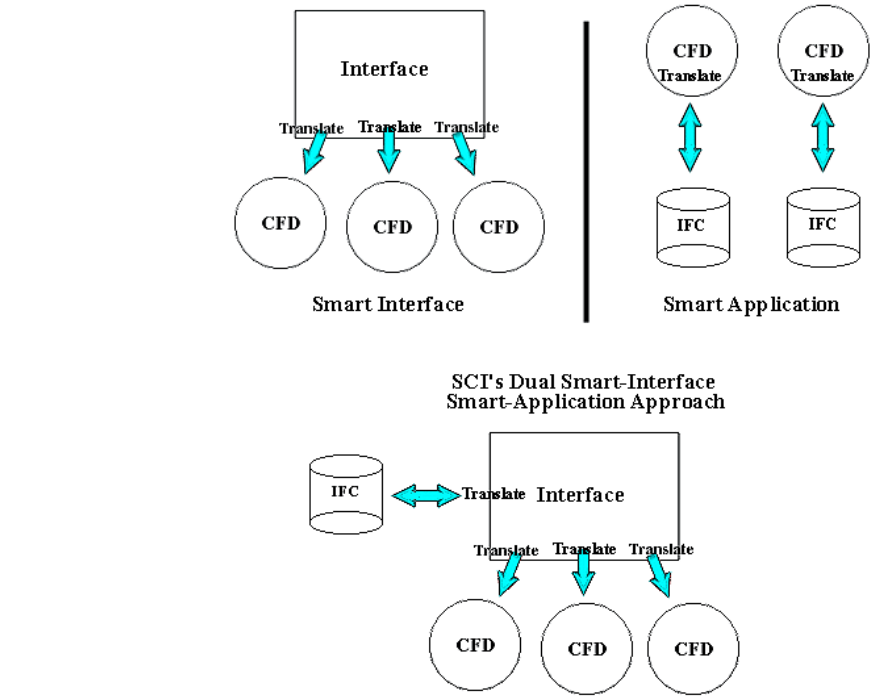

Figure 6: The difference between a smart interface and a smart application lies in where the model transformation occurs.

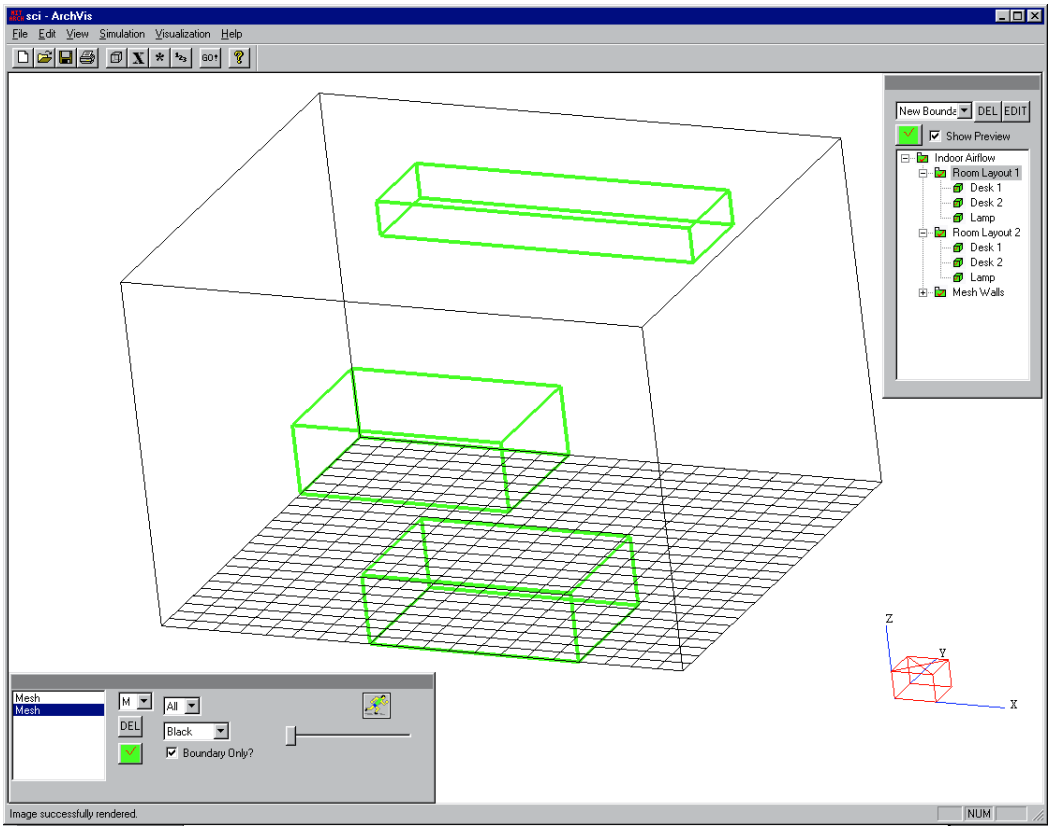

Figure 7: Two room layouts in the same SCI model, however the first room layout is disabled.

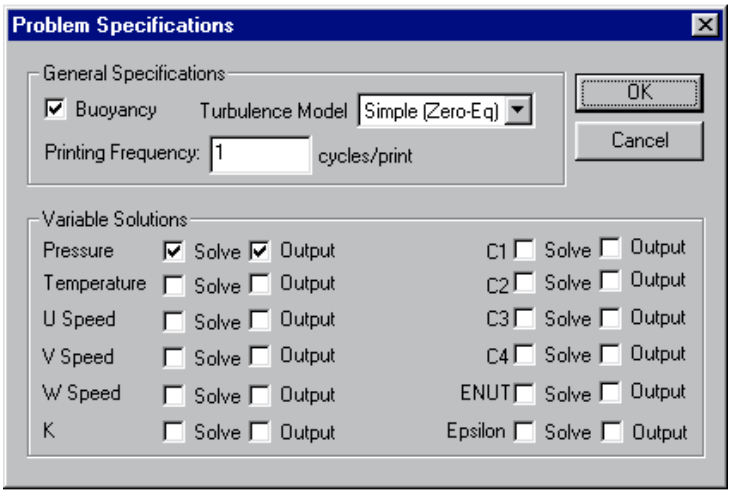

Figure 8: The problem specification tool.

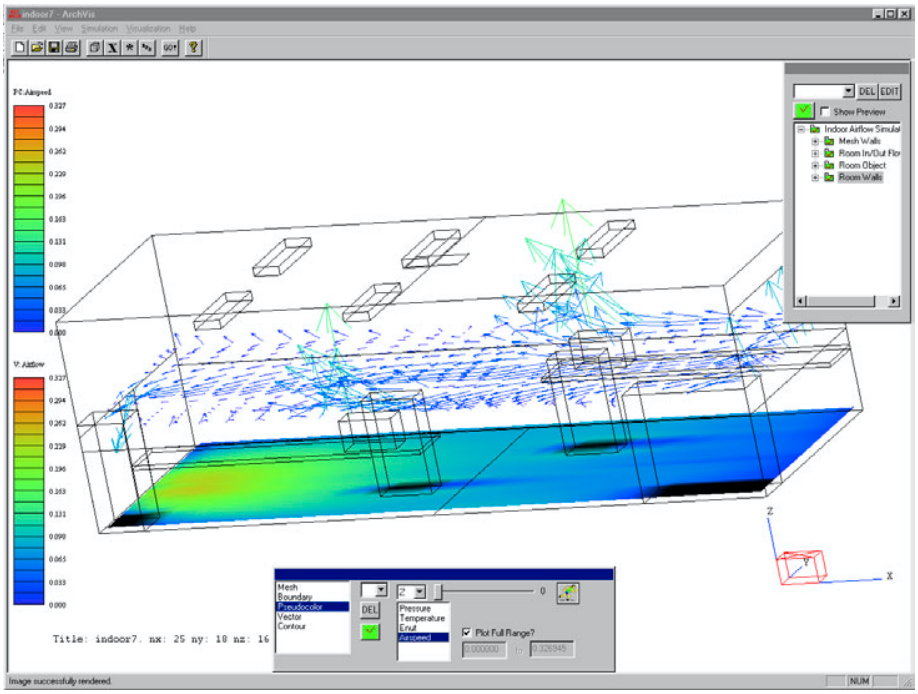

Figure 9: A vector and pseudocolor plot of airflow inside a room with two workstations.

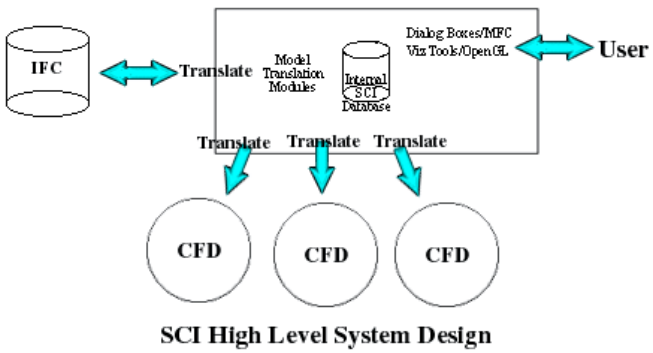

Figure 10: Looking at SCI's internal system design.