IFC-Based Framework for Evaluating Building Envelope Performance

Hua Sheng He

A Thesis

in

The Department of Building, Civil and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science (Building Engineering)
Concordia University
Montreal, Quebec, Canada

November 2005

Your file Votre référence
ISBN: 0-494-14237-5
Our file Notre référence
ISBN: 0-494-14237-5

Abstract

IFC-Based Framework for Evaluating Building Envelope Performance

Hua Sheng He

Building envelope is a subsystem in a housing system, but it is not a separate part of a house. Its performance and construction details should be taken into consideration with the house concurrently.

Computer simulation is one of the most significant methods to evaluate the performance of the building envelope. However, simulation programs are good for evaluating separate functions but do not address the global performance requirements for a building envelope system. Moreover, data input in some of these applications is complicated and time-consuming, and people who run the applications need advanced computer skills and professional experience. Furthermore, some of these applications cannot read the geometry data from CAD drawings, and this results in low efficiency and data input errors.

The objectives of the research study aims to develop an IFC (*Industry Foundation Classes*)-based framework to meet the requirements of evaluating building envelope performance by integrating evaluation applications and criteria with IT (Information Technology) and IFC. It investigates and compares applications and criteria, and then

develops a conceptual framework for integrating tools to evaluate building envelope performance and compare this performance against sets of criteria.

The framework extracts geometry and material layer data of a house from CAD drawings in IFC model, links to performance evaluation applications, such as HOT2000 and MOIST3.0, and compares evaluation results against a set of criteria. A prototype system has been developed including a preprocessor, an application integrator, and a postprocessor. Finally, a case study, which aims to validate this prototype system, is discussed.

Acknowledgements

First, I would like to gratefully thank my supervisor, Dr. Paul Fazio, who gave me the opportunity to pursue my Master's degree and provided the initial concept for this research project. He is nice, knowledgeable, and respectable and understands how to find out the advantages and potential capabilities of his students. Without his trust, understanding, encouragements, patience, and instructions, it is hard to imagine that I could succeed in this research study. In addition, I would also like to thank my co-supervisor, Dr. Amin Hammad, who has been willing to apply his knowledge of Information Technology (IT) to my research work in building engineering. He dedicated his efforts and time to guide my research in detail. Having worked together with him for one and half years, I have learnt from him not only how to do research, but also how to keep on moving forward to achieve goals.

In addition, I would like to thank Dr. Dominique Derome, who provided me with detailed instructions in the first part of my Master's study. Since my background is structural engineering, I had a hard time when I tried to shift my career into the building envelope. Dr. Derome kept on encouraging me. She enabled me confident enough to cope with the challenges in the new field.

Besides supervisors, many other experts have contributed to the success of this research study. I would like to express my gratitude to Mr. Jeff Blake and Mr. Brian Bradley from NRCan, who provided the API and Batch version of HOT2000; Mr. Patrick Houbaux and

Mr. Kari Karstila from Eurostep Group, who provided the IFC2x2 Active Toolbox evaluation version and implementation samples; Mr. Dino Gerbasi, who provided the data of the NOVTEC Advanced House; Dr. Hugues Rivard, who answered questions about IFC and provided information about CONDENSE; and Dr. Radu Zmeureanu, who answered questions about HOT2000.

During these demanding years of study, the support of my friends was extremely important. I would like to thank Dr. Jiwu Rao. He devoted his efforts and time to help me generate ideas, answer questions, search references, guide programming, and review drafts of papers and thesis. Moreover, I would like to thank Dr. Hua Ge, who answered my questions, discussed the project, and provided improved opinions to the project; Mr. Jinxing Zhang, a computer expert, who helped me use Microsoft.NET and answered my questions about software engineering; Mr. Khaled el Ammari, who shared his information about IFC with me, answered my questions, and helped me use the Architecture Desktop 2005; Ms. Miljana Horvat, who provided information about the evaluation criteria in her projects; Ms. Xiangjing Yang, who helped me rebuild the file of the NOVTEC Advanced House with HOT2000; and all other friends, who helped me in this project.

Finally, I would like to thank my family, my wife Yujuan, my children Qicong and Qiming, and my mother Lingxian for their love, support, and understanding. You are my motivation to pursue the graduate study. Without your contributions, it is impossible for me to succeed in this study. Furthermore, to Qicong and Qiming, I promise that the vacations I owed you in these years will be made up in the near future.

This project was being funded by Dr. Fazio's NSERC grant no. 4770/2002 and a grant from the Faculty of Engineering and Computer Science at Concordia University.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. General Background

Prefabricated house manufacturers and exporters in Canada identified the necessity and significance for developing performance evaluation tools. These tools would improve performances of housing systems, demonstrate the quality of prefabricated houses, and promote the house products in both domestic and international markets (Fazio and Poliquin, 2000; Horvat et al., 2001; Horvat et al., 2002). Moreover, they intend to reveal the deficiencies of the products and then correct them before the products are delivered. The existing performance evaluation programs or protocols in Canada and other countries have not only revealed a list of parameters to be evaluated but also created a set of criteria to be met before a housing system would be accepted (Horvat and Fazio, 2004).

The building envelope is a subsystem of a housing system. It functions as a barrier or filter to protect the occupants from cold and hot weathers, wind, rain, water vapor, solar radiation, outside noise, pollution, smoke and fire propagation, insects, and animals. Moreover, it must be structurally strong and stable, durable, aesthetically pleasing and economical (Hutcheon, 1963). Evaluating performances for building envelopes relates to all these multidisciplinary fields. In addition, even though the building envelope is a subsystem in a housing system, it is not a separate part when being designed; instead, its performance characteristics and construction details of various components should be taken into consideration concurrently (Fazio, 1990). Accordingly, a holistic approach of performance evaluation will not only ensure that numerous criteria of the building envelope, such as structure, thermal, integrity, moisture, acoustics, etc., are all satisfied, but also ensure that good integration is established between the envelope and the structure as well as the mechanical system of the building.

There are two categories of strategy to evaluate performances of buildings. One is evaluated by experimental tests, and the other is by computer simulations. Experimental tests could be carried out in laboratories or in fields but are more expensive and time-consuming. In contrast, computer simulations could be proceeded in personal computers thus need less money and time than experimental tests. At present, since computers are being widely used in building engineering, a large number of computer-based simulation applications are already available in the market, such as:

- HOT2000 (Buildings Group NRCan, 2005),
- EnergyPlus (U.S. Department of Energy, 2005),
- WUFI-ORNL/IBP (ORNL/IBP, 2005),
- hygIRC (IRC/NRC, 2005),
- MOIST3.0 (BFRL/NIST, 2005),
- CONDENSE (GES, 2005),
- and Airpak (Fluent Inc., 2005).

Although these applications are professional evaluation tools, they are not as widely used by frontline practitioners as developers expect. One reason is because these programs are good for evaluating separate functions but do not address the global performance requirements for a building envelope system subjected to climatic conditions of different regions of the world. Another reason is because data input in these applications is complicated and time-consuming, and people who run the applications need advanced computer skills and professional experiences. Moreover, some of these applications cannot read the geometry data, which are part of the input data from CAD drawings, and this results in low efficiency and data input errors (Suter et al., 2004). Accordingly, these applications are typically developed and used by researchers, and are rarely used in actual designs.

When evaluating the performance of the building envelope, building owners, manufacturers, designers, and evaluators need to know how a specific envelope system performs under a series of climatic loads for a specific region. They also need to know the performance requirements of the target location where the envelope system is built. Existing computer simulation programs address different characteristics of the building envelope such as energy consumption, resistance to heat loss, moisture movement, etc., but typically do not address the performance criteria for different regions of the world. Therefore, it is necessary to develop a set of rules (or clauses) as comparative standards or criteria for the evaluation of building performances in design and maintenance stages. At present, several programs or protocols in the world are already available to evaluate the overall or partial performance of the whole house. Based on the criteria developed in

Canada and the other countries, BEPAT (Building Envelope Performance Assessment Tool) is being developed at Concordia University (Horvat and Fazio, 2004). It integrates all codes about the building envelope including mechanical systems, structural systems, and energy conservation.

1.2. Integrated Framework with IT and International Standard

Computer technology has been widely used in building engineering. It is possible to evaluate building envelope performances in an integrated framework with Information Technology (IT) and international standards. This framework should have the capacity to extract the physical properties (e.g. dimensions, materials, etc.) from the Building Information Model (BIM) and then generate performance values (e.g. energy consumption) for a house in a given region. Moreover, the framework should compare the performance values against existing criteria for that region, which would be provided by the prevailing standards and codes. The framework should also have the capacity to link to application programs (e.g. HOT2000) to determine the characteristics (e.g. energy consumption) of a house. With new software engineering approaches, the framework will enable the user to easily input the characteristics and the target location of a given building envelope system. It provides building owners, manufacturers, designers, and evaluators with integrated methods and tools to evaluate the overall performance for the building envelope using multidisciplinary knowledge, the state-of-the-art IT, and the international standards IFC (*Industry Foundation Classes*). Its final result enables its users to evaluate the performance for the building envelope system by themselves without having to learn how to use each simulation program that is necessary for the evaluation.

1.3. Objectives

This research study presents an IFC-based framework to meet the requirements of evaluating building envelope performance by integrating evaluation applications and criteria with IT and IFC. The specific objectives of this research study are as the following:

(1) To investigate and compare applications and criteria used in evaluating the building envelope performance.

(2) To investigate a conceptual framework for integrating tools to evaluate building envelope performance and compare this performance against established criteria.

(3) To develop a proof-of-concept system to test and validate the conceptual framework.

Chapter 2

Literature Review

2.1. Introduction

Performance evaluation is significant and necessary for building envelopes both at the design and at the maintenance stages. There are many ways to implement this process, for example, the experimental test and computer simulation. The experimental test includes laboratory tests and field tests. Computer simulation includes the application of computer simulation programs, such as HOT2000, EnergyPlus, WUFI-ORNL/IBP, hygIRC, MOIST3.0, CONDENSE, and Airpak. Moreover, since evaluating the performances of building envelopes needs a set of criteria to identify whether the building envelope meets the requirements in a targeted climatic area, it is necessary to have a set of evaluation protocols integrated into the proposed framework.

The content of this chapter is part of the background and knowledge required for this research study. First, the method used to evaluate the building envelope performance by experimental test is briefly addressed. After that, several popular computer simulation applications are concisely reviewed. Moreover, a building envelope evaluation tool BEPAT, which is currently under development at Concordia University, and other evaluation criteria are shortly introduced. Furthermore, some code compliance checking research projects with knowledge-based systems are described. Finally, a building envelope evaluation framework for exterior envelope wall systems for the homebuilding industry in North America is presented.

2.2. Overview of Building Envelope and its Performance in a Building System

The building envelope is a subsystem in a building system. It serves as a barrier between the outdoor climate and the indoor environment and includes building components (e.g. the exterior walls, foundations, and roofs). They separate the interior from the exterior and form the building envelope subsystem. In addition to the building envelope subsystem, there are a number of other subsystems, such as heating, cooling, ventilating, domestic hot water supplying, and lighting, available in a building. All these subsystems not only work together with the building envelope but also influence the performances of the building envelope.

The building envelope is categorized in the building engineering domain. Currently, one of the most important concerns in building envelope research is to investigate the transport of HAM (Heart, Air, and Moisture) in and through the building envelope. This investigation is relevant to thermal performance, air tightness, and moisture management performance. Another important concern in building envelope research is its impact on energy consumption. According to Hydro Quebec, house characteristics in Canada are significant factors that directly influence energy consumption, especially in heating energy consumption. For a residential house, a building envelope with poor insulated external walls, roofs, and foundations could cause up to 40% of total heat loss; draft effect could cause up to 25% of total heat loss; and low quality doors and windows could cause up to 30% of total heat loss of a house (Hydro Quebec, 2005). Therefore, in terms of saving energy, providing building envelope with energy conservation characteristics is one of the most critical tasks for the decision makers who are concerned with the lifecycle cost of houses and with reducing energy consumption. By researching the response to the HAM transport and energy performance of the building envelope, it is possible to form the foundation for developing design guidelines for moisture control and management and energy conservation for the next generation of durable, affordable, and environmentally friendly buildings.

2.3. Building Envelope Performance Evaluation by the Experimental Tests

One significant strategy to evaluate the building envelope performance is by the experimental test such as full-scale laboratory and field tests. Usually, full-scale laboratory tests are used in buildings with innovative designs and new building systems, and field tests are used in buildings which are already built. Many full-scale laboratory tests have been implemented in the Environmental Chamber in Building Envelope Performance Laboratory in the Department of Building, Civil, and Environmental Engineering at Concordia University. One example of these tests is a research project: Moisture Occurrence in Roof Assemblies Containing Moisture Storing Insulation and Its Impact on the Durability of Building Envelope (Derome and Fazio, 1998; Derome, 2000). This research test was carried out in two full-scale test-huts in the Environmental Chamber and lasted 190 days. It investigated the patterns of moisture accumulation in single cavity flat roofs. These roofs were fully insulated with cellulose insulation which can store moisture within it. During the test, patterns of moisture movement and accumulation for a full wetting-drying cycle have been developed under the condition of daily cycle. Based on this project, a set of design guidelines were established for flat roof assemblies which consist of moisture storing materials such as cellulose. These guidelines have been used by the designer to evaluate the hygrothermal performance and the durability of a design house (Derome and Fazio, 1998; Derome, 2000). Another full-scale test example is also a research project: Impact of Added Insulation on the Hygrothermal Performance of Leaky Exterior Wall Assemblies (Desmarais et al. 2001; Desmarais, 2000). This research test was also carried out in a full-scale test-hut in the Environmental Chamber at Concordia University and lasted 119 days. The construction of the hut is insulated with glass fiber batt between the studs and was exposed to climatic conditions such as winter and late spring. This study proposed an experimental procedure to evaluate the effect of adding insulation to leak exterior wall assemblies with different air leakage paths (Desmarais et al. 2001; Desmarais, 2000). In addition to roofs and walls, other assemblies such as curtain walls can also be tested in full-scale. For example, a research project, Study on Overall Thermal Performance of Metal Curtain Walls (Ge et al., 2001; Ge, 2003), was also a full-scale test carried out in the same environmental chamber as the above two research projects. Although these full-scale tests can truly reflect the building envelope performance, they are expensive and time-consuming. This results in the development of another performance evaluation strategy, computer simulation.

2.4. Building Envelope Performance Evaluation by Computer Simulation

As computer technology has been widely applied in building engineering, computer modeling and simulation technology are playing more and more significant role in building performance evaluation. In contrast to the experimental test, computer simulation needs less money and time. By predicting the performances of buildings, computer simulation can help engineers to process non-trivial tasks, such as design and analysis, so as to provide the occupants with buildings which have less energy consumption, good indoor conditions, and less impact on the environment (Hensen et al., 1998). As a result, many researchers have contributed their effort to the development of computer modeling and simulation programs and have succeeded in developing many software programs, for example, HOT2000, EnergyPlus, WUFI-ORNL/IBP, hygIRC, MOIST3.0, CONDENSE, and Airpak.

2.4.1. General Introduction of Computer Simulation in Building Engineering

Building simulation has become an integral part in building engineering. Its primary objective is to conduct a performance analysis to determine a design decision, choose a dimension parameter, or estimate a budget allocation for maintenance. Although the current generation of simulation tools plays no significant role in the conceptual stage because decisions at this stage are mostly based on expertise and experiential knowledge of consultants, the proceedings of IBPSA (International Building Performance Simulation Association) conferences contain many publications which show more and more functions that have been offered by current building simulation applications. For example, significant progress has been made in areas such as performance prediction, optimization of system parameters, controls, etc. Accordingly, it may be expected that future trends in building simulation tools are primarily driven by the need of better design decisions and better quality control over the performance assessment process (Augenbroe, 2002). The following sections will introduce some computer simulation programs in building engineering.

2.4.2. CONDENSE

CONDENSE is a user-friendly simulation software in the AutoCAD environment. It is used to calculate the rate of condensation that occurs on the surface of the assemblies or the interstitial area between two layers. It can also specify the location of the dew point of water vapour diffusion in wall and roof. Moreover, it can calculate the R-value for each layer and the total R-value of the assembly so as to determinate the heat loss for the assembly. Furthermore, for projects in the Quebec area, it can estimate the approximate material costs for the assemblies.

In addition, CONDENSE has a graphical and menu-driven interface for data input. User can enter outdoor environmental conditions, such as location name, temperature, and relative humidity to initiate the analysis. It has 830 brand names and generic building products to be selected from its database. After the layers of the assembly are selected, CONDENSE will calculate the temperature and vapor pressure profiles against the cross section of the assembly and determine whether condensation will occur. If condensation occurs, it will calculate the dew-point temperature and the rate of condensation and illustrate the dew-point location in the cross section. The summary of the CONDENSE calculation includes: indication of condensation, dew-point temperature, rate of heat loss, rate of vapour diffusion, rates of concealed condensation and surface condensation, and cost of assembly (SIRICON, 2004; Rivard, 1993).

The simulation model of CONDENSE applied the dew point method in ASHRAE (1989). For the sake of simplification, it has many assumptions. For example, the R-value of the layer is not influenced by the moisture content; the flow of heat and water vapor are analyzed in 1D; indoor and outdoor environmental conditions are constant; the heat transfer and water vapor diffusion are considered in steady-state; and the air leakage is not considered in the analysis (SIRICON 2004; Rivard, 1993).

Although CONDENSE has been widely used in the building industry, especially in Quebec, it has the following limitations.

(1) It calculates the condensation, but cannot predict the relationship between the condensation and the deterioration in the assemblies.

(2) It analyzes the moisture transfer without considering the air leakage. When the assembly has air leakage, the amount of moisture transfer predicted by CONDENSE is not accurate because in winter the amount of condensation in the wall caused by indoor air exfiltration is far greater than that by diffusion.

(3) It does not consider the liquid capillary transport. Thus, the moisture transfer is underestimated when the moisture transfers in materials with high moisture content.

(4) It does not consider the effects of moisture and heat storage because of its steady state assumption. This may underestimate the potential damage in a wall and roof when they can store heat and moisture or when the weather cannot dry the assemblies rapidly.

(5) It does not consider the release and absorb of latent heat when moisture condenses or evaporates. Thus the temperature change occurred in the exposed surface is neglected.

(6) It analyzes only the assemblies in one-dimension. Therefore, it neglects the influence of thermal bridges such as effect of corner, holes, cracks, studs, etc. (SIRICON, 2004; Rivard, 1993).

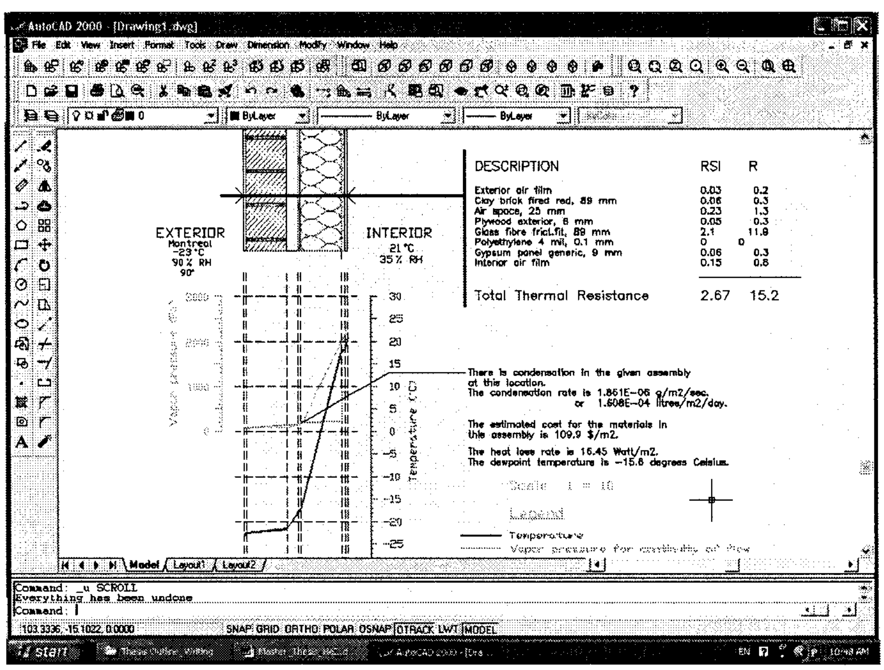


Figure 2-1. Presentation of Results by CONDENSE

CONDENSE may be used in assemblies of residential and commercial buildings. Its input data and output result are straightforward, and it takes only a little time to run the program. In some cases, it is more useful and efficient than other complicated software, for example, analyzing airtight construction and in case where wetting by rain or heating by direct sunlight does not play a significant role. In the conceptual design stage, it is a good option to use the CONDENSE because it is easy to compare different ideas in a short time. Figure 2-1 shows the presentation of results by CONDENSE.

2.4.3. WUFI ORNL/IBP

WUFI is an acronym for transient heat and moisture transport in German. It is designed to calculate the simultaneous heat and moisture transport in one-dimensional multi-layered building components. With a friendly user interface in the Windows environment, the user can build a project with a building assembly, calculate it, change the input date, and display the results in a graphical view. WUFI ORNL/IBP, one of WUFI's versions, is an easy-to use, menu-driven program for use on a personal computer that can provide customized solutions to moisture engineering and damage assessment for various building envelope systems such as walls, roofs, and basements exposed to natural climatic conditions. It is based on some building physics concepts such as sorption and suction isotherms, vapour diffusion, liquid transport, and phases changes (ORNL/IBP, 2005).

The input data in WUFI ORNL/IBP include: (1) The geometry, composition of the building assembly, and the numerical grid; (2) The hygrothermal properties of the relevant building materials; (3) The climatic indoor and outdoor boundary conditions and hourly weather data (e.g. temperature, relative humidity, wind, driving rain, and solar radiation); (4) The time steps and the numerical control parameters.

The output data in WUFI ORNL/IBP include: (1) Curves which describe the temporal evolution of certain quantities, taken at specified location or as mean values over specified layers; (2) Profiles which show the distribution of a quantity across the building component at a specified point in time; moreover, it automatically records the quantity profiles in the initial and final states; (3) A film which contains the profiles of all time steps. The user can save the calculation results together with the input data in a project file. It offers graphical functions that allow the use to view the computed courses and profiles, and to edit and print the graphs.

WUFI ORNL/IBP is a typical advance numerical model method. It calculates the simultaneous heat and moisture transport in one-dimensional multi-layered building components. In the calculation of heat transport, WUFI ORNL/IBP takes into account the thermal conduction, enthalpy flows through moisture movement with phase change, short-wave solar radiation, and nighttime long-wave radiation. However, convective heat transport by airflows has been disregarded, since it is usually difficult to quantify and is rarely one-dimensional. The vapour transport mechanisms included in WUFI are vapour diffusion and solution diffusion. However, convective vapour transport by airflows has been ignored. In addition, the liquid transport mechanisms take into account the capillary conduction and surface diffusion.

Like other simulation programs, WUFI ORNL/IBP also has many limitations:

(1) It deals only with 1D process and cannot treat 2D or 3D situations, such as thermal and moisture bridges.

(2) Transport phenomena, such as airflows in the component, uptake of ground water under hydrostatic pressure, and gravity effects, have been neglected. It cannot calculate extremely high-temperature ranges such as fire conditions. The interface between two capillary-active materials is treated as ideally conducting.

(3) The WUFI ORNL/IBP model currently employs an average absorption and desorption isotherm and does not account for hysteresis in its moisture storage function. The enthalpy flows resulting from the transport of liquid water across a temperature differential are ignored.

(4) It does not include time-dependent hygrothermal material properties. Contaminated surface due to exposure and cracks due to freeze thaw, swelling, and shrinkage are not accounted (Kuenzel, et al., 2001).

WUFI ORNL/IBP simulates the dynamic performance for the building envelope in the weather condition of a design year. Therefore, it takes time to run the program. Its input is not difficult since it has a user-friendly graphical interface. However, its output is much complicated, and it takes time to analyze. This program is suitable to be used for assemblies in both residential and commercial buildings. It can be used to estimate the drying time for masonry and lightweight structures with trapped or concealed construction moisture, investigate the danger of interstitial condensation, study the influence of driving rain on exterior building components, help to select repair and retrofit strategies, compare and rank different designs, and aid the development and optimization of innovative building materials and components. In contrast to CONDENSE, its simulation process is closer to the practice, and its output is more detailed. Figure 2-2 is one of the interfaces in WUFI ORNL/IBP.

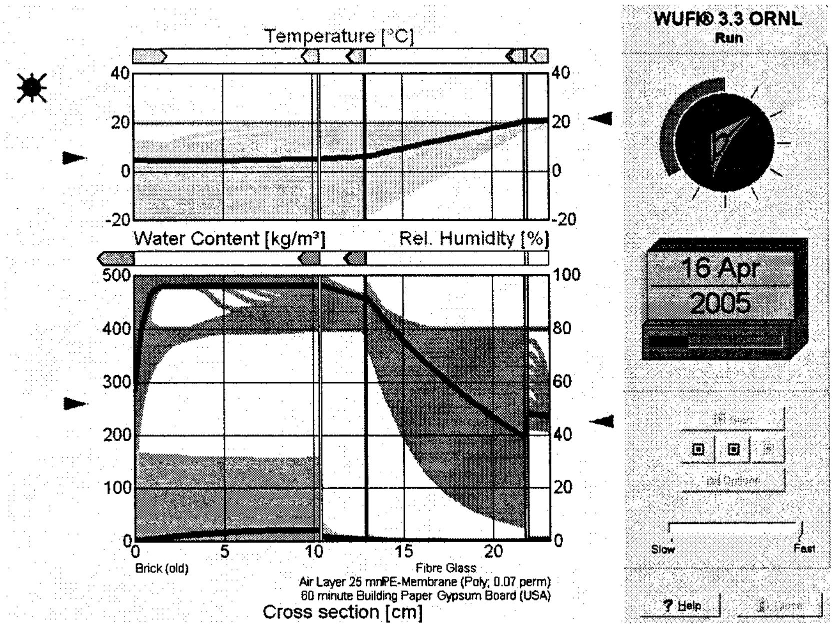


Figure 2-2. Animation of Thermal and Hygric Processes in WUFI ORNL/IBP

2.4.4. EnergyPlus

EnergyPlus is the official building simulation program of the United States Department of Energy. Its parent programs are BLAST (Building Loads Analysis and System Thermodynamics) and DOE-2 programs, both of which are energy analysis and thermal load simulation programs. The EnergyPlus's inputs are building's components and its mechanical systems, while its outputs are the heating and cooling loads necessary to maintain thermal control set points, conditions throughout a secondary HVAC system and coil loads, the energy consumption of primary plant equipment, and so on. EnergyPlus's input is based on ASCII text file, as shown in Figure 2-3, and its output is also based on ASCII text file. It is intended to be a simulation engine around which a third-party interface can be wrapped (U.S. Department of Energy, 2005). EnergyPlus is one of the important energy simulation programs that have implemented IFC (*Industry Foundation Classes*). More details about the application and research related to IFC in this program will be introduced in Chapter 3.

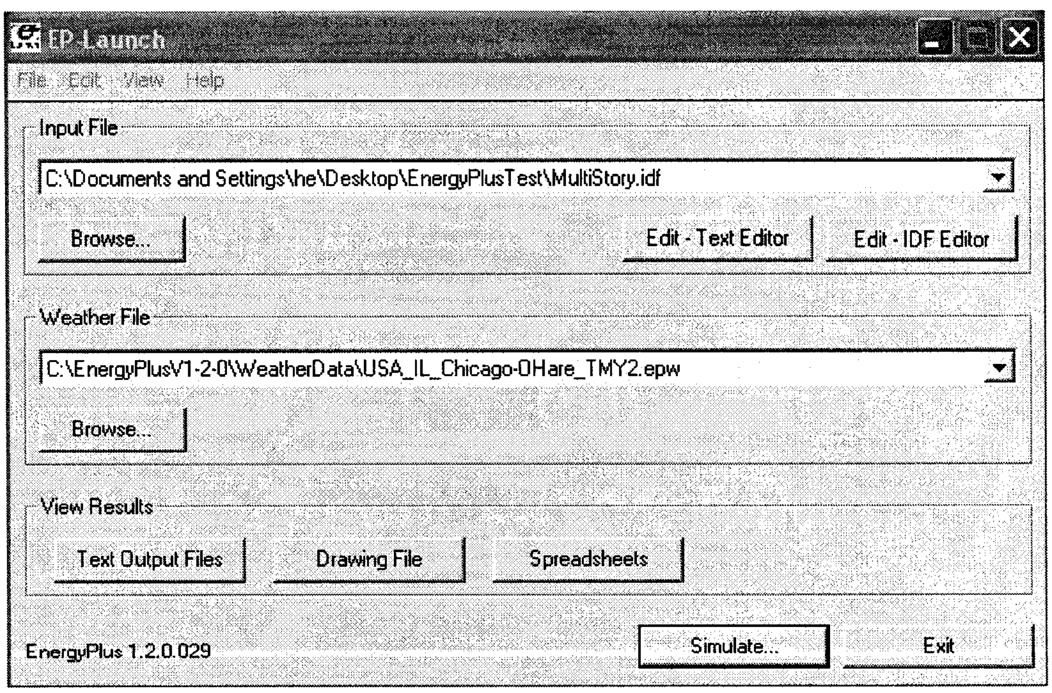


Figure 2-3. The EP-Launch Program in EnergyPlus

2.4.5. MOIST3.0

MOIST3.0 was developed by the National Institute of Standards and Technology (NIST) and was made available to the public in January 1988. It is a one-dimensional model which analyzes the heat and moisture transfer in wall, cathedral ceiling, and low slope roof constructions. MOIST3.0 is a user-friendly personal computer program.

MOIST3.0 has many possible applications, for example, predicting the wood decay by moisture contents, indicating the need of a vapour retarder in an assembly, determining the drying rates for construction materials, predicting mold and mildew growth by the surface relative humidity at the construction layers, determining the effect of moisture on thermal resistance, analyzing the effect of house tightness on wall moisture contents, assessing the effect of different wall construction materials or designs on wall moisture performance, investigating the effect of different indoor moisture generation, mechanical ventilation, or climate conditions on indoor relative humidity, investigating a wide range of applications for low-slope roofs and cathedral ceilings

Although MOIST3.0 has many possible applications, it still has limitations because of the assumptions of the model: (1) It does not consider the framing members and the vertical movement of moisture in an earth-coupled wall due to one-dimensional modeling; (2) It does not analyze the wetting mechanism in the exterior of the assemblies caused by rain; (3) It does not include the insulating effect and change in roof absorptance caused by snow on the roof; (4) It neglects the heat and moisture transfer caused by air movement due to the assumption of airtightness (Burch et al, 1997; Burch et al., 2001).

In this research, MOIST3.0 has been integrated in the framework because it has an Application Programming Interface (API). More details about MOIST3.0 will be presented in Chapter 5. Figures 2-4 and 2-5 are two sample interfaces about its input and output.

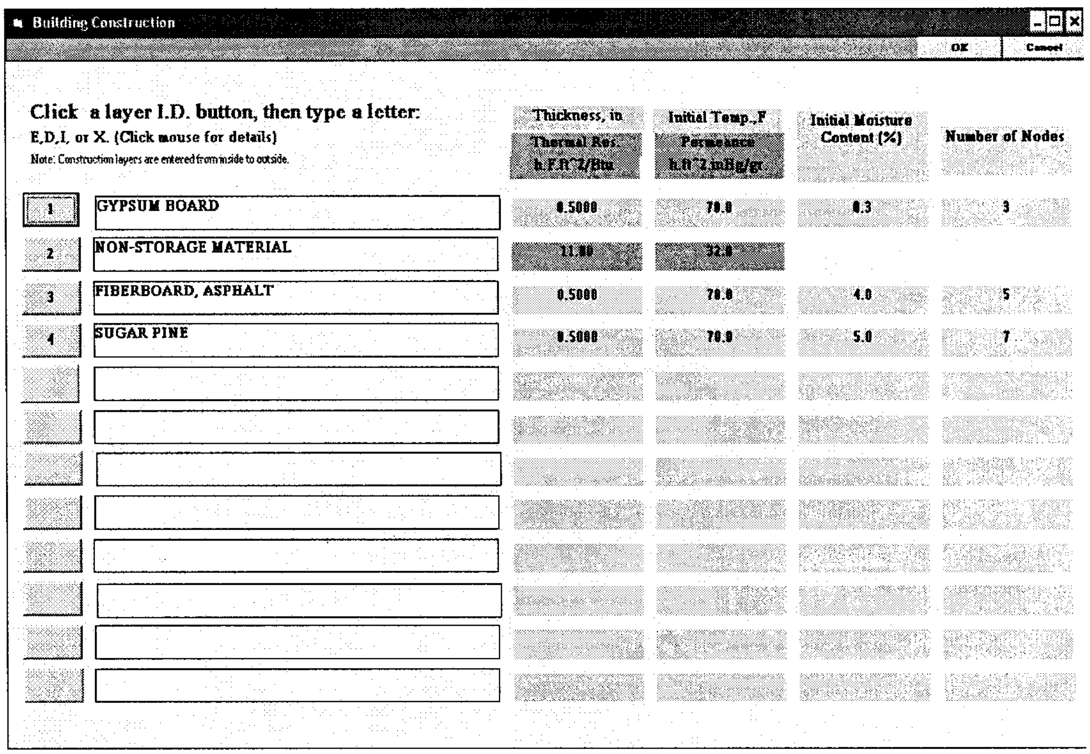


Figure 2-4. Building Construction Input Interface in MOIST3.0

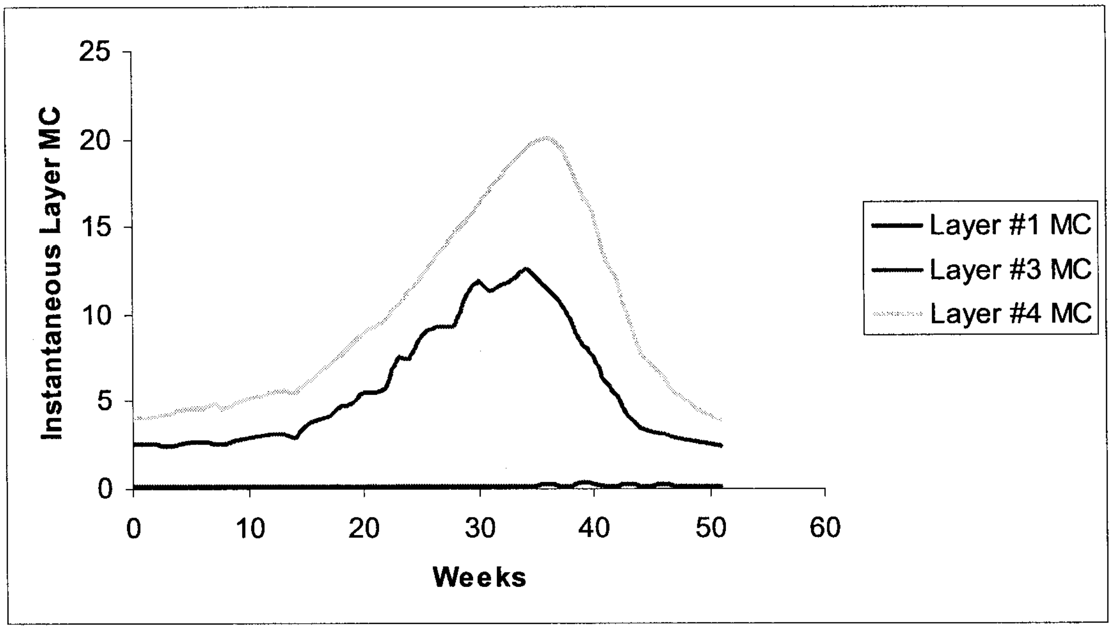


Figure 2-5. Layer MC Output in MOIST3.0

2.4.6. HOT2000

HOT2000 is a computer simulation tool that analyzes the energy consumption for low-rise residential buildings. Its latest version 9.2.1 was released in July 2004. It is the reference calculation program for the R-2000, which is a voluntary program for energy conservation in Canada. HOT2000 is supported by NRCan (Natural Resource Canada). It can evaluate the building accurately by its heat loss or gain and system performance models which includes the thermal effectiveness of the building and its components, the passive solar heating owing to the location of the building and the operation and performance of the building's ventilation, heating and cooling systems (Buildings Group/NRCan, 2005).

HOT2000 has a user-friendly GUI (Graphical User Interface). Its visual directory tree can conduct the user to input data for the specific ceiling, wall, floor, foundation, and HVAC system. Moreover, electrical appliances and hot water supply are taken into account as loads for energy consumption. In order to simplify the input for a new house files, HOT2000 provides the user with house template and wizard interface. These template and interface contain predefined files with default values, and the user can reuse data automatically. When calculating the space heating load and monthly and yearly energy consumption, it considers the utilized solar and internal gains. HOT2000 has a weather database for 75 cities in Canada, 200 cities in the U.S., and some selected cities in other countries.

HOT2000 is very powerful in terms of functionality and has many possible applications. For example, forecasting energy consumption, estimating energy costs and performance for natural gas, electric, propane, oil, and wood heating equipment, ensuring housing designs comply with energy code, calculating thermal resistance (R-value) of envelope components, improving the energy efficiency of building designs, predicting and controlling air infiltration, exploiting potential of passive solar heating, planning for adequate interior ventilation for good indoor air quality, and estimating energy consumption for space heating and cooling, water heating, lighting and appliances. Furthermore, the output of HOT2000 is very flexible and detailed. It consists of three kinds of reports, namely, technical report (including monthly tables), comparison report for comparing results of up to four houses simultaneously, and weather, fuel cost and economic reports (Buildings Group/NRCan, 1995; Buildings Group/NRCan, 2005).

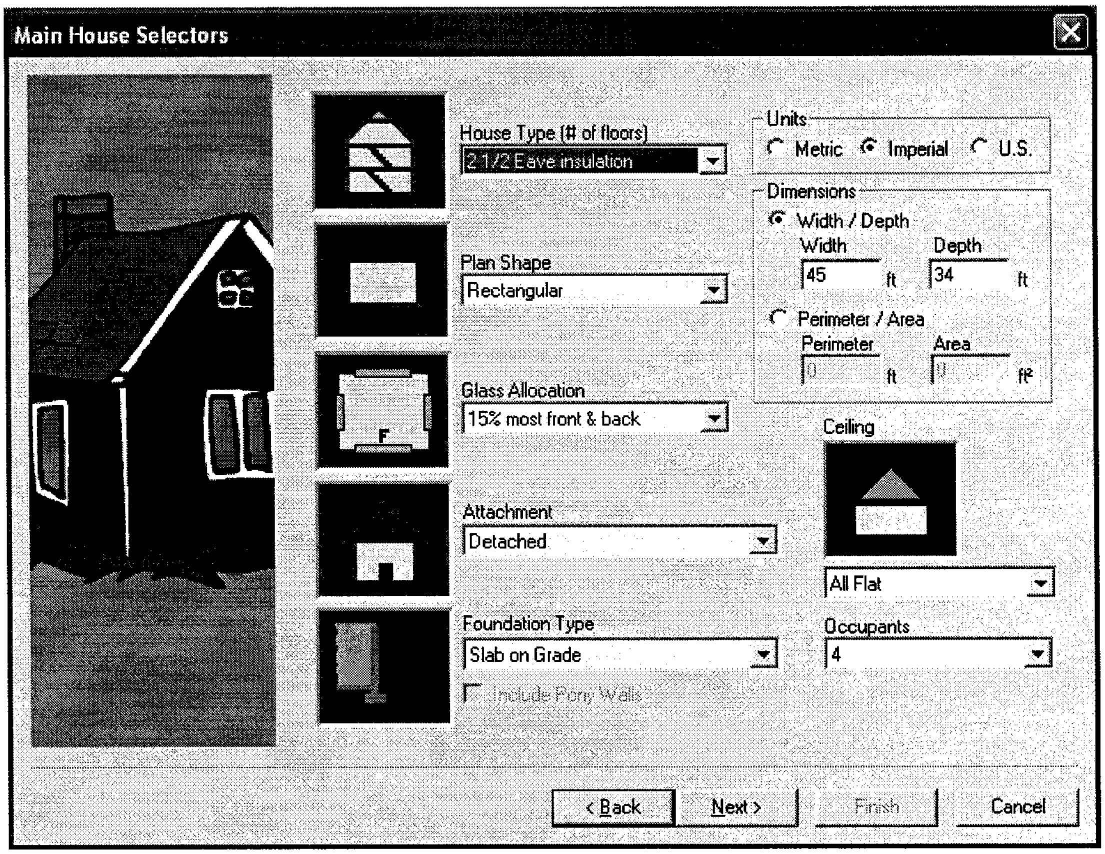


Figure 2-6. Main House Selectors Input Interface in HOT2000

Since the first release of the HOT2000, many researches have devoted their efforts in the application of the HOT2000. Baouendi (2003) and Baouendi et al. (2005) developed a prototype tool, EEE (Energy and Emission Estimator), for the evaluation of sustainability of Canadian houses. This prototype tool links to HOT2000 and estimates the life cycle energy uses, greenhouse gas emissions, and the cost of Canadian houses. It has a user-friendly interface and requires few inputs from its users. It imports the description of the house envelope and its annual energy consumption and cost directly from the HOT2000 program (Baouendi, 2003; Baouendi et al., 2005).

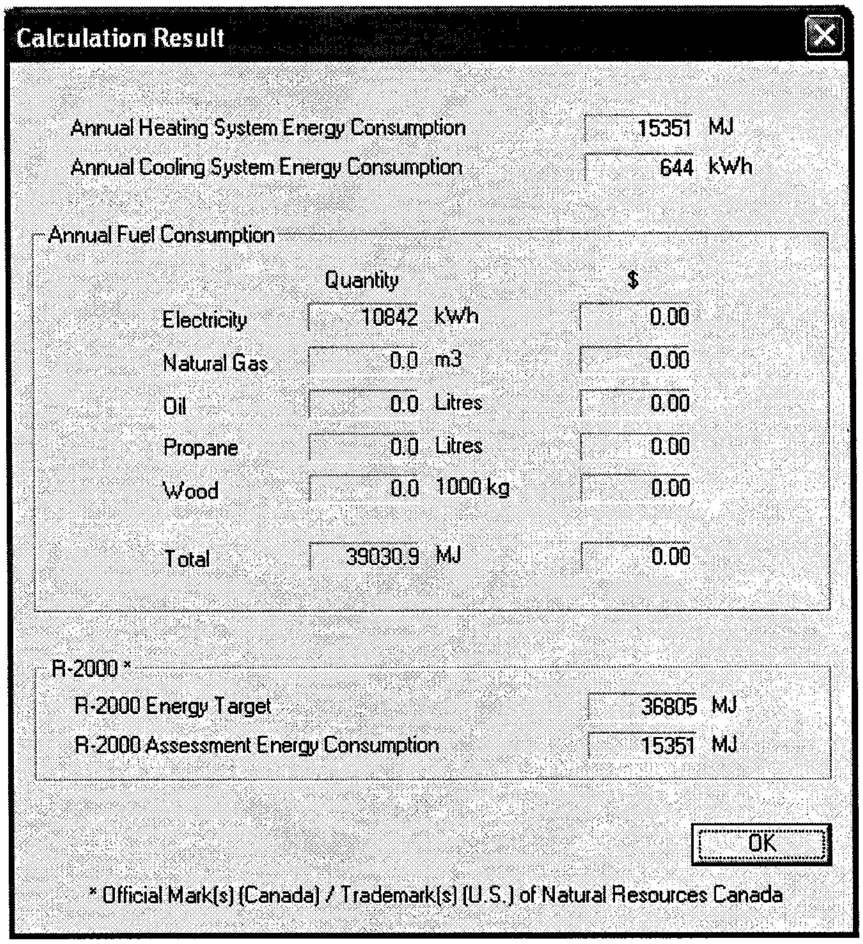


Figure 2-7. Calculation Output Interface in HOT2000

Like MOIST3.0, HOT2000 has been integrated in the proposed framework in this research study due to the availability of its API. Therefore, the above is just the brief introduction about HOT2000. More details will be presented in Chapter 5. Figures 2-6 and 2-7 are two sample interfaces of input and output in HOT2000.

2.5. Building Envelope Performance Evaluation Criteria

After obtaining the results from the performance evaluation programs, it is necessary to compare these results against a set of criteria to evaluate whether the results are within the established limits of relevant codes or guidelines. In fact, several programs or protocols in Canada and abroad have already been developed to evaluate the overall or part of the performances of a house. Some researchers, such as Miljana Horvat and Paul Fazio, members of the Building Envelope Group at Concordia University, Canada, have been doing research on integrating these protocols into a more comprehensive tool for building envelope performance evaluation (Horvat & Fazio, 2004). The following lists some examples of these protocols or programs. More of them could be found in a comparative review in Horvat and Fazio (2005).

(1) The P-mark from Sweden (SP, 2005),

(2) The Housing Quality Assurance Law (HQAL) from Japan (Eastin et al., 2000),

(3) The European Technical Approval Guidelines ETAG 007 from Europe (EOTA, 2002),

(4) The R-2000 program from Canada (NRCan, 2005),

(5) The Novoclimat from Quebec, Canada (Quebec Agency for Energy Efficiency, 2005),

(6) Leadership in Energy and Environmental Design (LEED) from USA (LEED Steering Committee, 2003).

To advance the concept of overall building performance, Miljana Horvat and Paul Fazio first undertook the classification of the requirements and standards which govern the building performances. This resulted in the Building Envelope Performance Assessment Tool (BEPAT) (Horvat & Fazio, 2004). The purpose of this tool is to evaluate the overall performance for light-frame site-built or prefabricated residential and small commercial building envelopes.

2.5.1. P-mark System for Prefabricated Houses

The P-mark system was developed by Statens Planverk (SP), Swedish National Board of Physical Planning and Building. It is a comprehensive quality assurance program and shows that the product and the quality control system employed in its manufacture have been examined and approved in accordance with current certification rules. The *P* symbol is not only a guarantee of quality to consumers but is also an indicator of good environmental practice in the use and manufacture of the products. The certification of the P-mark system for prefabricated detached houses including the following: requirements for certification, performance requirements for the finished house, quality system, quality plan, and SP's supervisory inspections. The performance requirements for the finished house include the airtightness, air exchange rate, airtightness of ducts, sound pressure levels, heat requirement and mean U-value, thermal comfort, and Radon concentration. The program covers all aspects of building envelope, including energy efficiency, which is controlled by insulation and air-tightness; durability, which is controlled by indoor air humidity; and the application of the rain screen principle to manage water penetration into the outer wall (SP, 2005).

2.5.2. European Technical Approval Guideline (ETAG) for Timber-frame Building Kits

The European Technical Approval Guideline (ETAG) was developed by the European Organization for Technical Approval (EOTA) which consists of eleven members such as Austria, Belgium, Finland, France, Germany, Italy, the Netherlands, Norway, Portugal, and the United Kingdom. In addition, it also includes corresponding members such as Demark, Iceland, and Slovenia. The ETAG provides performance requirements for timber frame building kits used in building construction, methods to evaluate performance, methods to assess performance for the intended use, and assumed conditions for the design and installation of the kits into a building. The ETAG covers the performance requirements of the building such as mechanical resistance and safety, safety in case of fire, hygiene, health and environment, safety in use, protection against noise, energy economy, and heat retention. The ETAG requires that durability is ensured by good design. Excessive moisture penetration caused by vapour diffusion, air leakage, or rain penetration should be prevented primarily by adequate construction details. The ETAG covers not only durability but also most other performance such as structural adequacy, fire performance, air quality, sound control, and energy performance (EOTA, 2002).

2.5.3. Standards for Evaluation Methods under the Housing Quality Assurance Law (HQAL) in Japan

The evaluation standards used by the HQAL were developed by the Ministry of Construction of Japan. The HQAL and its evaluation standards have been developed for single-family detached homes as well as multi-unit apartment buildings, with specific requirements made for concrete, steel, and wood-based building systems. It is based on the performance of the final product and applies equally to prefabricated and site-built housing systems. The general rules prescribe the performance evaluation included in two main stages, on the house design and on the house itself. The specific types of performance characteristics include the structural stability, fire safety, reduction of deterioration, consideration of maintenance, thermal environment, air quality, lighting (a visual environment), sound environment, and consideration for senior citizens and others special needs. HQAL do not deal with air leakage. Excessive moisture penetration in external walls is covered through the prescription of rain screens with ventilated air spaces between cladding and external sheathing. The HQAL is the most comprehensive quality assurance package in Japan (Eastin et al., 2000).

2.5.4. R-2000 Program

The R-2000 program consists of a set of performance-based standard. It specifies an energy consumption target for a house and a series of technical requirements for ventilation, airtightness, insulation, material selection, water use, and other factors which ensure that energy conservation is not achieved at the expense of occupant health or building component durability. The program was developed for both conventional builders and manufacturers of prefabricated homes. Technical requirements for R-2000 include: minimum envelope requirements, ventilation system requirements, combustion equipment requirements, energy performance targets, lights and appliances, indoor air quality, and environmental features. The R-2000 is a voluntary program. Despite its obvious merits for individual homeowners and the Canadian society, R-2000 has not attracted as many registrations as expected. The main obstacle to increasing the demand of R-2000 homes appears to be their high initial cost (NRCan, 2005).

2.5.5. Novoclimat

Novoclimat was recently developed by the Quebec Agency for Energy Efficiency. The program was inspired by the Canada's National Model Energy Code and the R-2000 program. It is a voluntary program. The main objectives of the program are to reduce thermal losses through better house insulation and increasing air tightness, maintain interior air quality through the selection of healthy materials and effective air distribution with fresh air intake, improve occupants' comfort, support thermal envelope durability, and introduce a building envelope quality control procedure based on infiltration measurements. The main categories of requirements are: thermal insulation, air-tightness (sealing of the thermal envelope), ventilation systems and equipments, indoor air quality and CO_2 content, space heating systems, domestic hot water heating systems, combustible heating appliances, water consumption, and lighting and other loads in winter time (Quebec Agency for Energy Efficiency, 2005).

2.5.6. LEED

LEED (Leadership in Energy and Environmental Design) was developed by U.S. Green Building Council in 2000. Its Green Building Rating System is the voluntary certification program that provides a complete framework for assessing building performance and meeting sustainability goals. LEED includes environmental aspects, such as sustainable sites, water efficiency, energy and atmosphere, materials and resources, and indoor environmental quality. In addition, it also consists of three principal requirements, such as prerequisites which list elements that must be fulfilled before a project can be considered for LEED certification, core credits which are given for meeting the requirements, and innovation credits which are extra credits given for exemplary performance beyond core credits or for innovative solutions that improve performance that are not covered otherwise in this rating system. Since the U.S. government accepted LEED, it has become popular and has been accepted in North America (LEED Steering Committee, 2003).

2.5.7. BEPAT (Building Envelope Performance Assessment Tool)

BEPAT (Building Envelope Performance Assessment Tool) is a framework that consists of codes, standards, guidelines, and criteria against which building envelopes can be evaluated. This research project is designed to evaluate the performance of residential or small commercial light-framed building envelopes which include site-built and prefabricated systems. The potential users could be architects, building envelope designers, home builders and producers of factory-made houses. BEPAT evaluates the functional requirements for building envelope such as air tightness, moisture management performance, thermal performance, energy performance, structural stability, acoustic performance, and fire control. This protocol can assess the building envelope from the design stage to the execution and installation phase. It has been implemented in Microsoft Excel to perform some internal calculations. Moreover, when using this tool to evaluate some functional requirements of the building envelope such as energy consumption, users should generate data by models, calculation methods, and computer programs, such as HOT2000, CONDENSE, MOIST3.0, etc. Then, the tool can use these data to compare with the benchmark values. Furthermore, this tool has a scoring and a weighting system. The scoring system assigns points which range from -2, -1, 0, +1, +2, +3 according to the performance evaluated against the specific criterion; the weighting system reflects the priorities among the parameters. The priority can be changed depending on the users' need. At present, the development of the criteria is based on Montreal's climatic, technical, and social environment (Horvat and Fazio, 2004). Figure 2-8 is the BEPAT in Excel sheet.

PEPAT is a comprehensive evaluation tool. It is developed based on other evaluation programs or protocols in Canada and other courtiers. Therefore, this research study applies it as the criteria to evaluate the performances of the building envelopes.

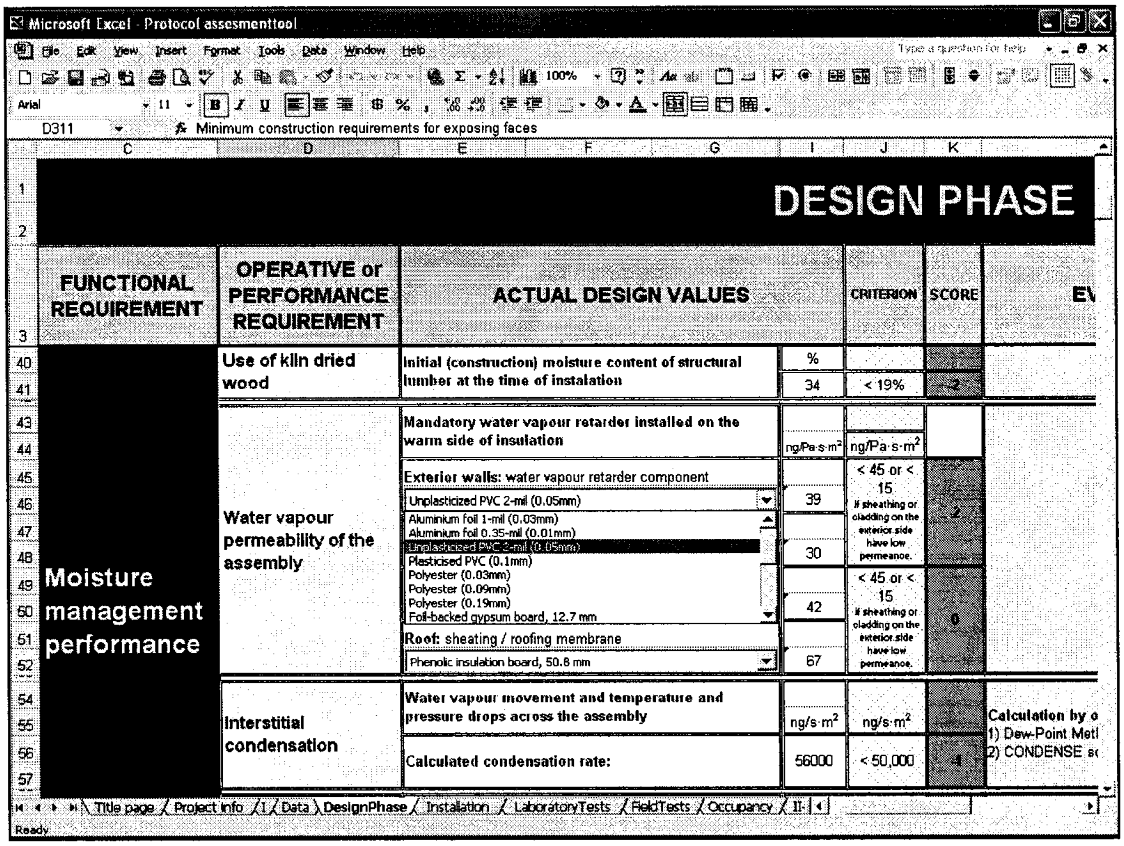


Figure 2-8. BEPAT in Excel Format (from Horvat & Fazio, 2004)

2.6. Code Compliance Checking for Energy Code with Knowledge-based Systems

When comparing the evaluation results with the criteria, code compliance checking is needed. In this case, a knowledge-based system such as an expert system, which is a computer program that performs a task normally done by an expert or consultant, is necessary. Expert systems are interactive computer programs incorporating judgment, rules of thumb, intuition, and other expertise to provide knowledgeable advice about a variety of tasks (Dym and Levitt, 1991). A research project titled A Knowledge-based System for Energy-Efficient Building Design (Gu, 1999) was one example to demonstrate the concept of code compliance checking. In this project, a systematic and integrated methodology was proposed to assemble a computer-based system that would perform as a design assistant. It helped architects and building designers at the preliminary design stage to compare the designs to the MNECCB (Model National Energy Code of Canada for Buildings) in areas of building envelope and lighting. A prototype system named Building Energy Code Advisor (BECA) was implemented in the AutoCAD environment in Autolisp [1995] with the integration of Level 5 Object [1994]. Autolisp is used to program the interactive interface that collects design data from building drawings. Level 5 Object is a knowledge-based expert system development tool that can utilize a database within a rule-processing procedure in an objected-oriented programming environment (Gu, 1999). Another research project titled Knowledge-based System Approach to Building Envelope Design (Gowri, 1990) also demonstrated the concept of compliance checking for building envelope. In this research, a prototype system known as BEADS (Building Envelope Analysis and Design System) was developed. It could generate feasible design alternatives, rank these alternatives, and select the best suitable one for external walls, roof and glazings in a building envelope system at the preliminary design stage (Gowri, 1990).

2.7. Integration of Building Envelope Evaluation

Since the building envelope has become more important in building engineering, many researchers have dedicated their efforts to apply integration method to evaluate the building envelopes. One example is Mund (2002). The research project, entitled

Evaluation Framework for Building Envelope, developed an evaluation framework for four exterior envelope wall systems: AAC (Autoclaved Aerated Concrete), ICFs (Insulated Concrete Forms), SIP (Structural Insulated Panels), and Integra Wall System. The framework was divided into a part for homebuyer evaluations and a part for homebuilder evaluations (Mund, 2002). Another example is a research project titled Towards the Holistic Assessment of Building Performance Based on an Integrated Simulation Approach (Citherlet, 2001). This project proposed an approach that incorporated different performances within a single program. The efforts undertaken in this work focused on the design of the building data model, the implementation of a single application, and the application of a case study to assess the building performance such as thermal, lighting, acoustics, occupant comfort, and the environmental impacts generated by the building during its whole life span. The model developed in this work has been implemented into ESP-r (ESRU, 2005), which is an existing simulation application that is capable of modeling transient energy and fluid flows within combined building and plant systems. In order to achieve a holistic approach, its capabilities have been extended to support the assessment of room acoustics and the life cycle impact (Citherlet, 2001).

2.8. Summary of the Literature Review

The experimental test and computer simulation are two main strategies to evaluate building envelopes. Although the experimental test is more reliable than computer simulation in terms of evaluation results, it is more expensive and time-consuming.

Computer simulation, on the other hand, needs less time and effort. With more applications of Information Technology (IT) in building engineering, it can be predicted that computer simulation will play an increasing role in the field. In addition, evaluation of the building envelope needs the assistance of computer-based simulation programs and a set of evaluation criteria. Due to the previous efforts of many researchers, these programs and criteria are already available in the field. Based on the simulation programs, evaluation criteria, and other researchers' studies reviewed in this chapter, it is reasonable and achievable to develop an integrated framework for evaluating the performances of building envelopes.

Chapter 3

IFC and its Applications in Building Engineering

3.1. Introduction

The research project presented in this thesis proposes an integrated framework which is based on the international standard, *Industry Foundation Classes* (IFC). The purpose of this standard is to make interoperability possible among different applications or even among different platforms. IFC is specific to the AEC/FM (Architecture, Engineering, Construction /Facilities Management). It is an open resource and is easy to get technical support from the Web. Since IFC is still being developed, more updated releases will be coming in the future. This chapter presents the IFC and its current status: the concept and history of IFC, the application of IFC in energy simulation and code compliance checking, etc.

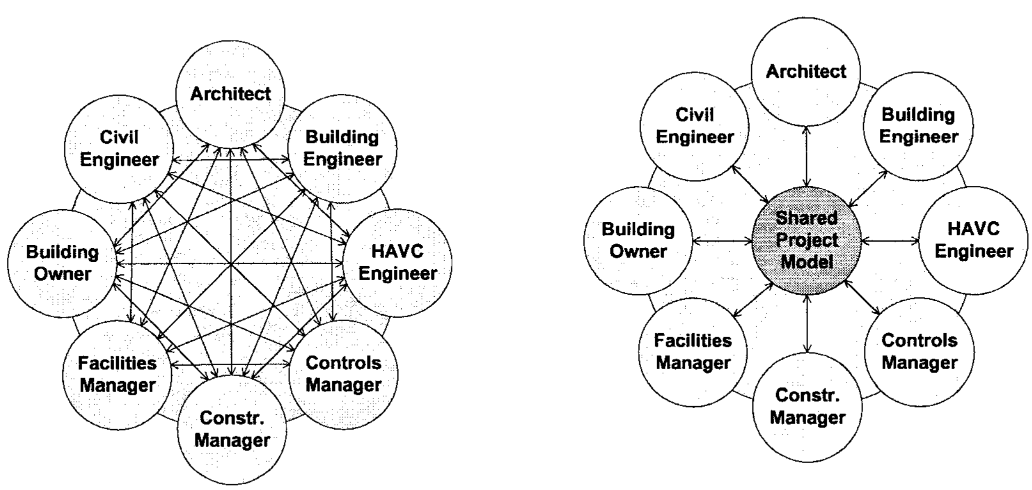
Information Technology (IT) has been widely used in AEC/FM, such as performance evaluation, CAD drawing, structural analysis, HVAC design, cost estimation, construction management, and code compliance checking. When practitioners use software applications, they need to transfer and share data among different applications developed by different software developers. However, interoperability among these applications is difficult to achieve since they are not developed based on the same data format.

IFC is developed by the IAI (International Alliance for Interoperability) (IAI, 2005a) as an international standard for building data sharing and exchange. These data are AEC/FM objects including physical objects, e.g., walls, and abstract objects, e.g. projects. Based on the EXPRESS language, which is used in ISO STEP project (ISO, 1999), IFC data are organized into a hierarchy structure with the object-oriented method. Furthermore, it is free, open and available to all AEC/FM software developers. Thus, interoperability among AEC/FM software applications is achievable.

Since the first IFC version released, many software developers have developed computer tools compatible with IFC, for example, Architecture Desktop and ArchiCAD in CAD drawing; Robot with Robin Building Modeler (ISS, 2005) in structural design; EnergyPlus (US Department of Energy, 2005) and RIUSKA (Granlund, 2005) in energy simulation; AirPack (Fluent Inc., 2005) in indoor air quality control; Timberline office (Timberline, 2005) in cost estimation. Meanwhile, many researchers have carried out their research work on the implementation of IFC in building engineering. For example, Romberg et al. (2004) used geometric models based on IFC for the finite element analysis in structural design. Other examples could be shown in energy simulation and code checking (Hammad et al., 2005).

3.2. The Concept of the Neutral Layer

The application of computers in AEC/FM has provided remarkable benefit over paper drawing since it is accurate, easy to modify, and convenient to archive. However, practitioners working in different fields of building engineering usually apply different computer applications separately. They cannot share or exchange data with each other so as to reduce the manual re-entry of design data. For the sake of interoperability, it is necessary to have a neutral layer to achieve this goal. Figure 3-1 shows the difference of data exchange or sharing between direct communication (e.g. CAD model) and neutral format (e.g. IFC Model). Actually, some neutral layers or information standards have been created, for example, Initial Graphics Exchange Specification (IGES), Standard for the Exchange of Product Model Data (STEP), and *Industry Foundation Classes* (IFC). The IGES is the first generation of the neutral layer. After that, STEP was developed into an international standard by ISO based on IGES. IFC, also an international standard, is being developed by the IAI, based on the same schema language EXPRESS, ISO 10303 Part 11 (ISO, 1994) and physical clear text format, ISO 10303 Part 21 (ISO, 2002) as that of STEP.



CAD Model **IFC Model**

Figure 3-1. Data Exchanging and Sharing in CAD Model and IFC Model (From VTT, 2005)

3.3. Initial Graphics Exchange Specification (IGES)

The IGES was the first national standard for CAD data exchange in U.S.A. It was created in 1979 by the CAD vendor community (Goldstein et al., 1998). It may be used as a 2D or 3D model in a CAD system and serves as a neutral data format to transfer the design drawing from one CAD application to another. With the IGES, design drawings are exported to IGES files which could be exchanged and imported into other applications (IGES, 2005). An IGES file is in ASCII clear text format and consists of several sections such as directory entry and parameter data. The IGES file includes a large number of entities. Each entity consists of a directory entry and a parameter data entry. The directory entries include indexes and descriptive attributes about the data. They are organized in fixed fields. The parameter data provides the specific entity definition. They are variable in length and format (Smith, 1988).

3.4. The Standard for the Exchange of Product Model Data (STEP)

STEP is a comprehensive ISO standard, ISO 10303, which describes how to represent and exchange digital product information. Since digital product data must cover the entire lifecycle of a product, STEP includes geometry, topology, tolerances, relationships, attributes, assemblies, configuration, etc. It has been constructed as a multi-part ISO standard. The basic parts are complete and published while more are under development. These parts cover general areas, such as testing procedures, file formats, programming interfaces, and industry specific information. The STEP is extensible because it is built on the EXPRESS language which can describe the structure and correctness of engineering information. EXPRESS is used to represent detailed information which is required to describe products of different industries. The Application Protocols developed for different industries in STEP are the main part of this standard and are the basis for STEP product data exchange. In addition, the EXPRESS language can document constraints and data structures. These formal constraints are an explicit correctness standard for the digital product data. Figure 3-2 shows the structure of the STEP standard. Infrastructure parts, such as the exchange file format (Part 21), have been separated from the industry specific information models, such as the application protocol for configuration-controlled designs (Part 203, also called AP-203). Most of the infrastructure is complete, but industry-specific protocols are open-ended (STEP Tools, 2005).

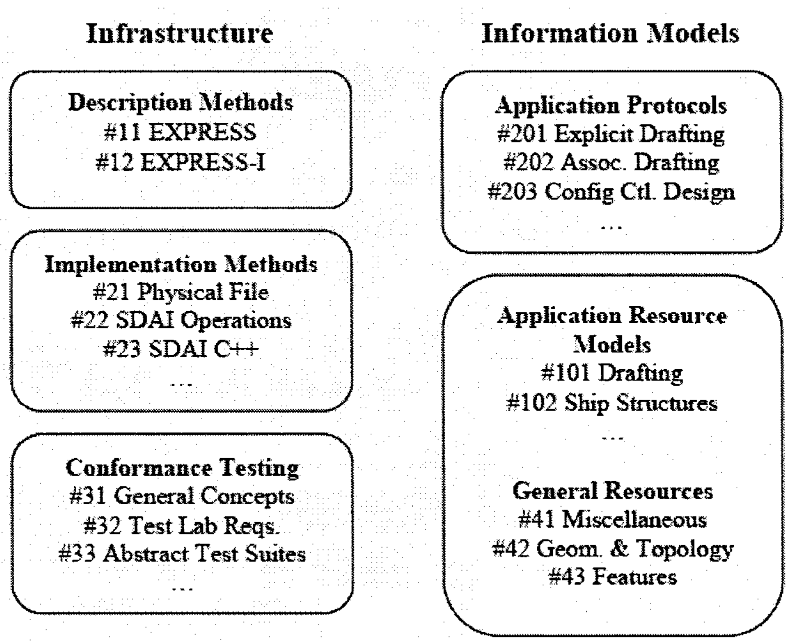


Figure 3-2. The Structure of the STEP Standard (STEP Tools, 2005)

3.5. *Industry Foundation Classes* (IFC)

IFC is an international standard for AEC/FM industries which is developed by IAI based on STEP. It was established in 1995 by American and European AEC/FM firms to promote interoperability in the industry. The alliance now has chapters in Europe, Japan, Korea, Singapore, Australasia, and America with over 650 member companies and has made interoperability a reality for AEC/FM firms.

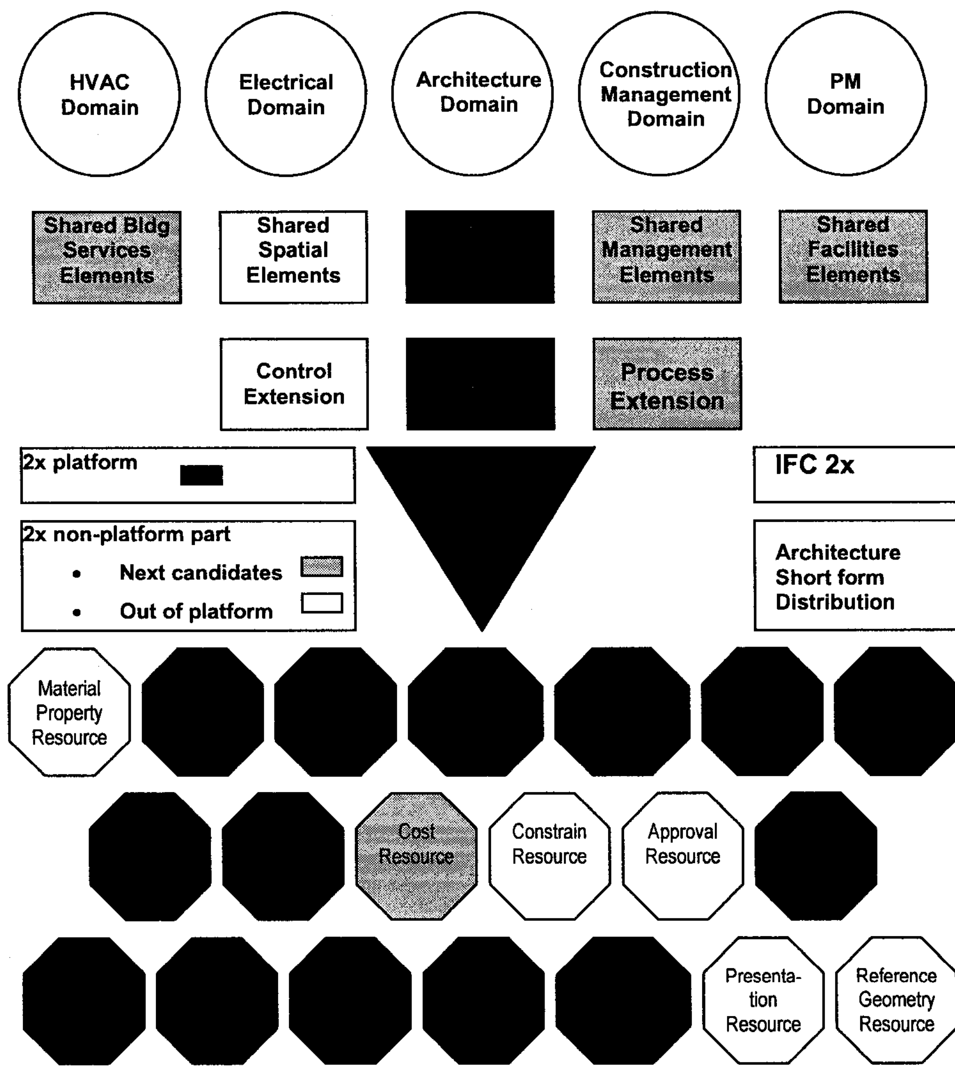


Figure 3-3. Schema Overview of IFC2x (IAI, 2005e)

IFC is developed in an extensible structure. Figure 3-3 shows the schema overview of IFC2x version. It consists of several layers such as Resource, Kernel, Extension, Shared Elements, and Domain. Moreover, it can be breakdown into two parts: 2x platform and 2x non-platform. The 2x platform (e.g., Kernel, Product Extension, Shared Building Elements, and most of the Resources) is the stable part, but the 2x non-platform (e.g., all Domains, some Shared Elements, some Extensions, and some Resources) might be changed within a future edition of the IFC2x model.

One feature of IFC is its 3D model. Unlike traditional techniques such as 2D files in DWG (DraWinG) /DXF (Data eXchange File), which are graphics file formats created by Autodesk and supported by CAD product, IFC provides not only a full geometric description in 3D but also the location and relationships as well as all the properties of each object, such as finish, serial number, and material description (Graphisoft, 2005). The concept object in IFC can be a physical object such as a door or an abstract object such as construction cost. In traditional 2D CAD environment, the software deals with entities such as line, arc, and text but not the objects directly. Software compatible with IFC model, on the other hand, deals with 3D objects directly. They reside in an integrated environment and have meaningful names such as IfcWall, IfcSlab, and IfcRoof.

Another feature of IFC is its interoperability. It enables the exchange of information between two dissimilar software applications. The user can use data in his/her application from another project participant on a different system and vice versa. However, traditional CAD software is based on a drafting paradigm. The users could not access data completely and sometimes not at all because there was no standard for defining the parts of a building. It is almost impossible for the non-CAD users to access information for costing, construction management, etc. Software developers with IFC, on the other hand, can create universal applications by incorporating AEC/FM objects. Applications that support IFC will allow members of a project team to share project data in an electronic format. This will ensure that the data is consistent and coordinated. Furthermore, the data can be shared throughout the whole lifecycle of the building. That is, information generated by the project design team will be available in an intelligent, electronic format to the building construction team, and facilities managers through their IFC compliant software.

The most important feature of IFC is its object-oriented approach. The IAI has specified the 'things' that could occur in a constructed facilities including physical objects such as doors, walls, fans, etc. and abstract objects such as space, organization, processes, etc. These specifications represent a data structure supporting an electronic project model useful in sharing data across applications. They are called classes which describe a range of components that have common characteristics. For instance, every door has a characteristic of opening to allow entry to a space; every window has the characteristic of transparency so that it can be seen through. *Door* and *Window* are names of classes. These classes defined by the IAI are termed *Industry Foundation Classes*. As in computer science, when the concept of a class is applied, each instance of the class is called an object (IAI, 2005d).

The IAI has successfully included industry leaders from all aspects of the AEC/FM community. According to the development structure of IFC in Figure 3-4, the IAI members have participated in defining the general requirements (e.g. IFC DEFINITION in Figure 3-4) for data exchange throughout the project life cycle. A number of IFC data models have been released and a large number of software vendors (e.g. IMPLEMENTATION in Figure 3-4) have already adopted these standards in their products. However, the IAI is not a software development organization. It works independently from the software companies that serve the AEC/FM industry, but by developing the IFC specifications, the IAI enables these software vendors to create a new generation of software applications based on a standard IFC Object Model (e.g. IFC MODEL EXCHANGE in Figure 3-4) that apply the potential of IT in the AEC/FM industry. This common project model serves as the foundation upon which software authors can develop applications that read and write physical file formats used to communicate with the IFC (IAI, 2005d).

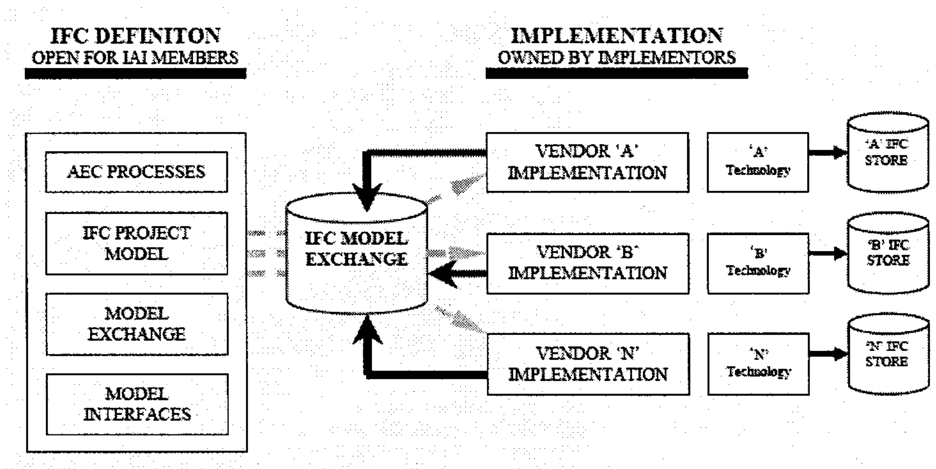


Figure 3-4. The Set up for IFC Development (IAI, 2005d)

3.6. The EXPRESS Language

IFC is developed based on STEP and applies the STEP's schema language EXPRESS (ISO 10303-11) as its description method. One of the objectives of STEP is to provide a neutral mechanism which is capable of describing product data throughout the life cycle of a product. This neutral mechanism is independent of any hardware or software (ISO 10303-11, 1994). This nature makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving data. EXPRESS is a description method that can help to achieve this objective. EXPRESS is used to specify the information requirements of other parts of the ISO-10303. It is a conceptual schema language which provides for the specification of classes belonging to a defined domain, the information or attributes pertaining to those classes (colour, size, shape etc.), and the constraints on those classes (unique, exclusive etc.). It is also used to define the relations, which exist between classes and the constraints applying to such relations (O' Sullivan et al., 2002).

Based on the same concept as in STEP, one of the most important functions of EXPRESS in IFC is to define classes. Within the class definition, all the attributes and behaviors, which characterize a class, are declared. The class is declared by the keyword ENTITY and terminated by the keyword END_ENTITY, as shown in Figure 3-5. An entity declaration creates a class and gives it a name. Attributes, such as Identifier, OwningApp, and OwningUser, are the characteristics (data or behavior) which are required to support use and understanding of the class. Attributes may be represented by simple data types, such as IfcString and IfcActor. Moreover, it can be represented by other classes (IAI, 2005b).

Another method to represent EXPRESS language is EXPRESS-G. It is a graphical modeling notation and is also developed within STEP. Like EXPRESS, EXPRESS-G is also used to identify classes, the data attributes of classes, and the relationships that exist between classes. Figure 3-6 shows the EXPRESS-G language definition for a supertype/subtype relationship. For the layered element, wall, floor, and roof slab have already been indicated as subtypes. Each subtype has all the characteristics of the layered element acquired by inheritance. However, each subtype may have additional attributes. To make the subtypes exclusive, that is, an IfcLayeredElement may be an IfcWall or an IfcFloor or an IfcRoofSlab, the number, "1", is written at the branch of the relation. The term ABS is used with the IfcLayeredElement to indicate that it is an abstract supertype. This means that it cannot exist by itself and only by virtue of its subtypes (ISO 10303-11, 1994; IAI, 2005c).

```
ENTITY IfcOwnerID;
Identifier: IfcString;
OwningApp : IfcString;
OwningUser : IfcActor;
END_ENTITY;
```

Figure 3-5. The EXPRESS Language Definition for a Class

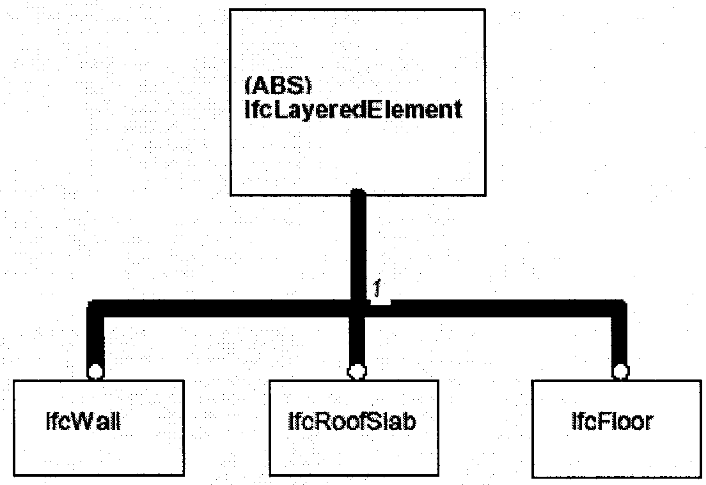


Figure 3-6. The EXPRESS-G Language Definition for a Supertype/Subtype Relationship

3.7. IFC Clear Text Format

The physical format of an IFC file is in clear text format. This is why data in IFC model can be transferred and exchanged between different software and hardware. Like EXPRESS, IFC uses the same clear text file format as that in STEP. Its implementation method is described in ISO 10303-21. The following example is a segment code in an IFC text file which expresses material layers in a wall.

Example:

```
#31 = IFCMATERIAL ('Gypsum Board');
#32 = IFCMATERIALLAYER (#31, 0.5., $);
#33 = IFCMATERIAL ('Glass Fiber Batts');
#34 = IFCMATERIALLAYER (#33, 4.0, $);
```

#35 = IFCMATERIAL ('Brick');

#36 = IFCMATERIALLAYER (#35, 3.5, $);

#37 = IFCMATERIALLAYERSET ((#32, #34, #36), 'Standard');

Explanation:

(1) Each line in the file expresses an IFC entity;

(2) The format of each entity in each line is:

Number of Entity = The Name of the Entity (Attribute $_1$, Attribute $_2$, ..., Attribute $_n$);

(3) The attribute list (*Attribute $_1$, Attribute $_2$, ..., Attribute $_n$*) contains the values of the attribute (e.g. the name of the IfcMaterial in #35 is 'Brick'), or the number of another entity (e.g. the material in IfcMaterialLayer #36 is "#35");

(4) The number #31, #32, #33, #34, #35, #36, and #37 are the numbers of the entities in the IFC file;

(5) Line #35 means that the material is brick;

(6) Line #36 means that the material layer is made of brick, which is the same as in line #35, and its thickness is 3.5 inch;

(7) Line #37 means that the wall is composed of three layers, which are the same as in line #32, #34, and #36.

3.8. Toolbox - Middleware to Access (Import or Export) IFC File

Most of the new architectural CAD tools, such as Autodesk (Autodesk, 2005), Graphisoft (Graphisoft, 2005) and Nemetschek (Nemetschek, 2005), are already IFC-compatible.

They can transfer CAD drawings into IFC text files automatically. Due to the complexity of the IFC object model, it is very difficult for a third-party application to read the huge amount of IFC data and extract them directly. In order to make this work easier, it is necessary to develop a middleware tool, called toolbox, to enable the software developers to achieve IFC compatibility with a reasonable amount of work. At present, there are several toolboxes available which are developed by some IAI members, for example, Eurostep, Olof Granlund Oy-BSPro COM Server, EPM-EXPRESS Data Manager, and Secom Ltd's Yoshinobu Adachi-IFCsvr ActiveX Component and IFC Model Server. Figure 3-7 shows the role of a toolbox in IFC implementation. There are three layers in this figure: the application, IFC toolbox, and IFC product data. The application's IFC pre/post processor reads and writes the IFC product data by the middleware IFC toolbox. The introduction of the Eurostep Active Toolbox, which is used in this proposed research study, is shown in Appendix A.

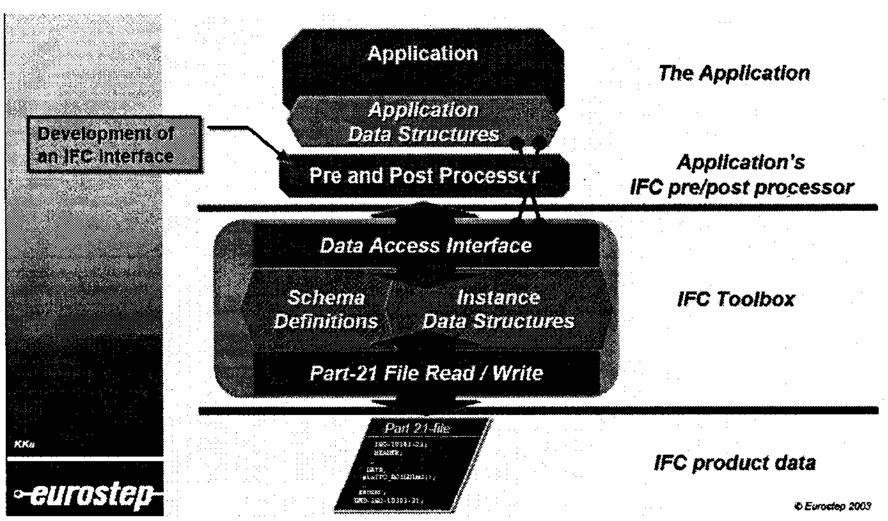


Figure 3-7. The Principal of Toolbox in IFC implementation (from Eurostep Group, 2005)

3.9. Application of IFC in Energy Simulation

EnergyPlus is one of the most important energy simulation software that has implemented IFC. The main component that is related to IFC is its IFCtoIDF utility, which extracts the geometric representation of building, space, and envelope surface and opening object instances contained within an IFC data file. It creates an EnergyPlus IDF (Input Data File) containing these objects mapped to the EnergyPlus Input Data Dictionary (IDD).

IFCtoIDF utility is a Window Dynamic Link Library (DLL). It reads an IFC data file, maps the geometry of relevant object instances to EnergyPlus objects, and writes an EnergyPlus IDF. However, this is not an interactive process in which the user can intervene to make changes or additions to the resulting IDF. The utility is built upon, and requires the installation of, the BSPro-COM-Server software package developed by Olof Granlund Oy in Finland (BSPRO, 2005). It is an IFC toolbox that provides methods for extracting data from files compliant with Releases 1.5.1, 2.0, 2x, and 2x2 of the IFC data model.

Although the IFCtoIDF utility has made the EnergyPlus compatible with IFC, it still has many limitations. There are no guarantees that it will produce accurate and consistent results when applied to IFC data files generated by different tools. There is not yet consistent support of user input for thermal zoning information in currently available IFC compliant tools. The IFCtoIDF cannot extract material characteristics of building envelope elements from an existing IFC data file. Since it runs essentially in a batch mode, no user interaction is involved. Modifications to the resulting IDF must be accomplished using some other interfaces such as a standard text editor or a specialized IDF editor (U.S. Department of Energy, 2005).

In addition to the IFCtoIDF, many researchers have published their literature about the application of IFC in EnergyPlus. Bazjanac (2004) discussed a new IFC HVAC extension schema, which is integrated in the latest IFC2x2. The development of this extension was part of the IAI BS (Building Service)-8 project series. It properly models two important general modeling features: connectivity, which includes any type of connecting HVAC equipment and parts, and time series, which enables the modeling of data that are defined dynamically and change in time. In the above reference (Bazjanac, 2004), the author takes a one-story bank branch building as an example to demonstrate the usage of IFC. As shown in Figure 3-8, a 3D building model is defined in ArchiCAD and saved in IFC format, the EnergyPlus then imports the building geometry data from the IFC file by the BS-Pro-COM Server, a middleware or toolbox that simplifies complex geometry definitions into simpler form as needed by non-CAD tools. In this case, the geometry data of the building is imported into EnergyPlus by the BS-Pro-Client. These data are organized in the form which is needed specifically by EnergyPlus. Meanwhile, data about HVAC system is entered manually according to the EnergyPlus definitions. The design-day simulation in EnergyPlus determined the size of the system. It also generated the zonal air-flow data, which were then exported directly to MagiCAD (MagiCAD, 2005) by the BS-Pro-Clients. The MagiCAD is an AutoCAD based program. It can draw and calculate the ventilation, heating, water, and sewage systems. In this project, it checked air-flow rates in devices, sized and balanced ducts (both supply and exhaust), and returned the results of calculation, the duct pressure, back to EnergyPlus by the BS-Pro-Clients. The EnergyPlus then calculated a new annual energy consumption which included imported duct pressure data from MagiCAD.

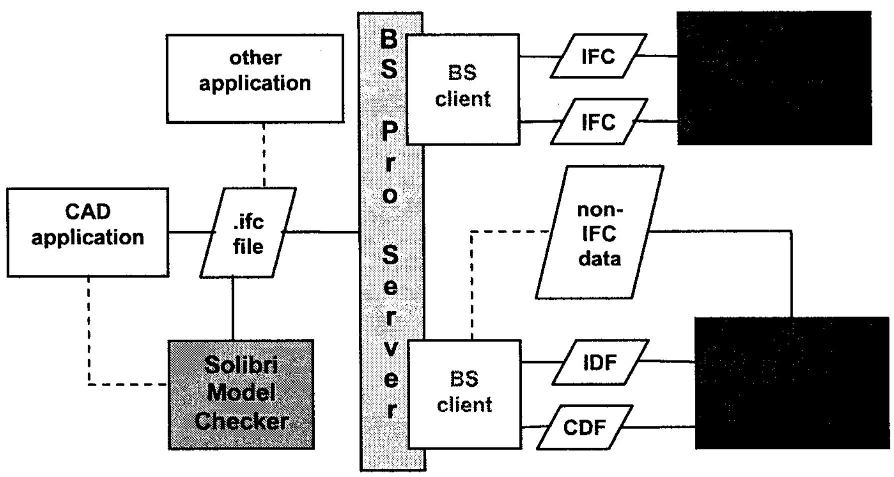


Figure 3-8. Data Paths between CAD and Applications by BS Pro COM Server (From Bazjanac, 2004)

In addition, Bazjanac & Maile (2004) introduced a new IFC HVAC Interface to EnergyPlus which was released at the end of 2003 and is used to convert HVAC data in IFC2x2 into corresponding definitions in EnergyPlus IDF. It can also be used to convert and add HVAC data contained in EnergyPlus 1.1.0, 1.1.1 and 1.2 data files to IFC2x2 data files. The development of this interface was also part of the IAI BS-8 project series. It provides two major functionalities: data conversion from IFC to IDF format, and vice versa. In both conversion processes, conversion is limited to data that directly describe HVAC components and systems in EnergyPlus. All other data, such as those defining various schedules (even if they pertain to the defined HVAC components and systems), are ignored. To successfully convert data from IFC to IDF and vice versa, IDF data must be consistent with IDF syntax. To assure that, the internal data structure of the interface is based on EnergyPlus IDD that defines all objects in EnergyPlus; all conversions to and from IDF format strictly follow definitions and rules defined in the IDD file associated with (and part of) the particular version of EnergyPlus. The complete process, which demonstrates geometry data from IFC to IDF by IFCtoIDF utility and geometry and HVAC data from IDF to IFC by IFC HVAC interface to EnergyPlus, is shown in Figure 3-9.

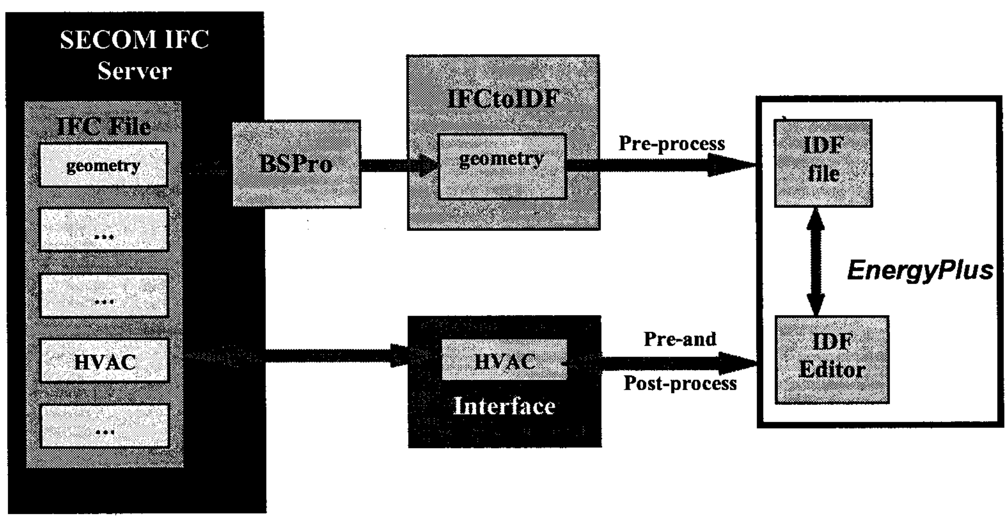


Figure 3-9. IFC HVAC Interface to EnergyPlus (From Bazjanac & Maile, 2004)

Earlier instance of energy simulation with IFC could be shown in RIUSKA. It is a tool for the dynamic simulation of comfort and energy consumption in building services design and facilities management. The RIUSKA simulates thermal conditions such as heating and cooling for individual spaces. It can be used to compare the dimension of the HVAC systems and to calculate the energy consumption for a building. The core of this simulation tool is DOE 2.1 E. RIUSKA has been certified as compatible with the IFC 1.5.1, 2.0 and 2x standards (Granlund, 2005). It has an AutoCAD-based space modeling tool called SMOG (Space Modeler by Olof Granlund Oy) and can create 3D objects of walls, windows, doors and spaces. Moreover, it can create connections between these objects. RIUSKA could import and export IFC files and transfer building geometry and construction data from IFC-compliant architectural software into other applications such as energy and comfort simulation, visualization, and life cycle cost analysis. Also, results calculated in RIUSKA can be exported to other IFC-compliant software (Karola et al., 1999).

3.10. Application of IFC in Code Compliance Checking

In code compliance checking, CORENET (COnstruction and Real Estate NETwork) in Singapore has developed a code-checking approach based on IFC. Their Integrated Building Plan and Integrated Building Services (IBP/IBS) e-Plan Checking will enable the government to approve building plans submitted by architects and building services engineers on the Internet. Automatic plan checking requires the submission of an IFC2x file. After submittal, the files are stored in a multi-user database at the plan-checking server, and all relevant authorities are notified. Each inspector can access the building plan, view and comment on it, and run an automatic check against the clauses from the building code relevant to his or her agency. The building code concerned for automatic plan checking includes provisions for area and volumetric extent calculations, overall thermal performance, fire escape, fire compartmentalization, disabled access, parking facilities, housing shelters, waste and drainage, sprinkler systems, fire fighting equipment, etc. (Liebich et al., 2002). This project may be extended to offer performance-based code checking. A model-mapping engine called eThermal has been developed within the novaCITYNETS ePlanCheck framework (FORNAXTM) to accomplish the task of translating CAD models to EnergyPlus model for thermal and energy simulation (Wong, 2004).

The following is a clause analysis example (Liebich et al., 2002). It demonstrates how the IFC is used in code checking.

Clause:

No exit, exit staircase or other exit facilities shall be narrower than the minimum width requirement as specified under Table 2.2A. The minimum clear width of an exit door opening shall be not less than 850mm (The clause 2.2.7 from the "means of escape" building regulation stipulated in Singapore).

Analysis:

Step 1. Assess all information that is required to establish the input for code checking. That is, what constitutes the "clear width" (as shown in Figure 3-10)?

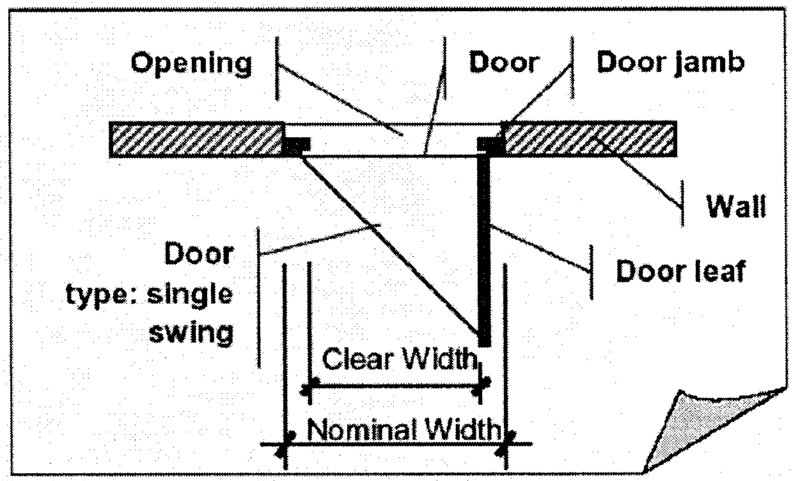


Figure 3-10. Clause Assessment

Step 2. Match the required input information to the entities available in IFC2x. In this clause, all information could be matched with the entities in IFC2x schema (as shown in Figure 3-11).

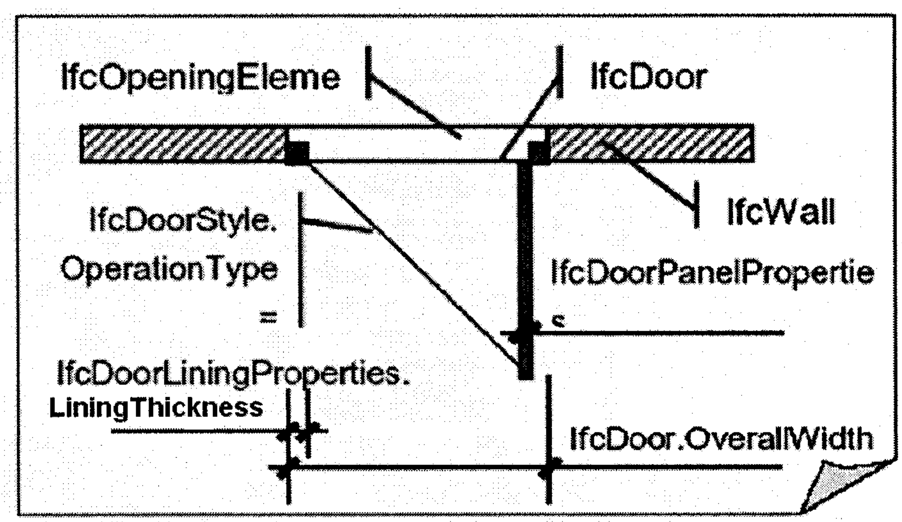


Figure 3-11. Underline of IFC 2x Model

Step 3. Calculate the clear width from IFC2x Information.

Case 1: if *IfcDoorStyle.OperationType =SingleSwing* then *OverallWith – (2*LiningThickness + 1 * PanelThickness);*

Case 2: if *IfcDoorStyle.OperationType =DoubleSwing* Then *OverallWith – (2*LiningThickness + 2*PanelThickness).*

Step 4. Compare the calculated clear width with the value in the clause:

Case 1: if *ClearWidth* > = *MinimumWidth* then Passed;

Case 2: if *ClearWidth* < *MinimumWidth* then Failed

3.11. IFC Extension Schema Development

IFC is an evolving international standard and can be extended in the future. With its various extension schema, IFC could include objects in AEC/FM which are not included in the current IFC version. Normally, extensions are developed by extension projects with guidance and quality assurance from IAI technical groups and officers. After the implementers finish the extension work, the extension models will feed into the next IFC platform release in no less than two to three years (IAI, 2005a).

The CORENET project (Liebich et al., 2002) is an example of IFC extensions. As part of the overall project, each clause of the building regulation, which is selected to be in scope of the e-Plan Checking, is analyzed for its information input requirements. The required input is then compared with the information provided by the IFC2x product model (as the clause analysis example in the section 3.10). If the information requirements are not directly reflected by IFC2x or cannot be generated from other information within the model, they will be dealt with within the IFC extension project Code and Standards (CAS) and will be added to the IFC2x edition 2. This extension is fully coordinated with other parts of the IFC 2x2, such as IAI BS8 project, which is related to building services, and IAI EL1 project, which is related to electrical services. This guarantees that the effort in developing a data model in CORENET project could be accepted internationally (Liebich et al., 2002).

3.12. Summary of IFC and its Applications

Applying international standard such as IFC is the new trend in building simulation. IFC is one of the best Building Information Model (BIM) since it is an open resource standard and has been accepted widely by researchers, developers, architects, and engineers in the building simulation community. Based on the great amount of previous research study, the latest IFC release 2x2 schema, and its extension projects, it is rational to develop an IFC-based framework to evaluate the building envelope performances.

Chapter 4

Building Envelope Evaluation Framework

4.1. Introduction

The project presented in this thesis is an IFC-based framework to evaluate the performances of the building envelope subsystem. It provides building owners, manufacturers, designers, and evaluators with integrated methods and tools to evaluate the performance of the building envelope using multidisciplinary knowledge, state-of-the-art IT, and the international standard IFC. With the application of IFC, The framework can not only import data from CAD drawing and exchange data between the preprocessor and the simulation programs, but also interoperate with other frameworks and software using standard data format. Its final result enables its users to do a preliminary evaluation of the performance of the building envelope system by themselves without having to learn how to use each of the simulation programs that are necessary for the evaluation. Engineers and other professionals with appropriate knowledge of building envelope issues could then carry out the evaluation without having to use individual simulation programs and manually match the results with the specific performance criteria. An overall rating would be obtained and, where this rating falls below expectations, corrective measures would be readily applied. This project would extend the knowledge base in the area of the building envelope and in the application of software engineering to the field of building engineering (He et. al., 2005).

This chapter presents the concept of the proposed integrated framework for the building envelope performance evaluation. It is currently developed for the building envelope, but it can be extended to evaluate a complete building. The flowchart of the framework is shown in Figure 4-1. It consists of three components: preprocessor, application integrator, and postprocessor. A CAD application, which is compatible with IFC, is also an important element that relates to the framework. The following sections will introduce the functionalities for each component and discuss some relevant concepts that have been included in the framework.

4.2. The Features of the Proposed Framework

The features of the proposed evaluation framework are:

(1) Extracting the physical properties, such as dimensions, materials, etc., from the CAD drawings in IFC model and then transferring these data into the simulation programs.

(2) Integrating simulation programs to enable the user to easily input the characteristics of a given building envelope system and its target location.

(3) Linking to application software, such as HOT2000, to determine the performance value, such as the energy consumption of a house for a given region.

(4) Comparing this performance value against existing criteria for that region which would be provided by the prevailing standards and codes.

(5) Interoperating, if necessary, with other systems and software by the IFC model.

4.3. The Structure of the Evaluation Framework

The overview of this Framework is shown is Figure 4-1 and Figure 4-2. The Framework consists of the following components and user interfaces.

(1) An IFC-compatible CAD application. It is a CAD tool which can transfer 2D or 3D CAD drawing, files in DraWinG (DWG) or Data eXchange File (DXF), into IFC files. Meanwhile, it can also read IFC files and transfer them into 2D or 3D CAD drawings. The AutoDesk's Architectural Desktop with its IFC2x utility and Graphisoft's ArchiCAD with its add-on interface are two typical IFC-compatible CAD applications available in the market.

(2) A preprocessor. It is a functional unit in the framework that processes the input data for the building envelope. It is used to access an IFC file from the CAD software; input data from end-users; store data to, and retrieve data from, the database; and build a middle layer (internal data structure) to transfer data to its receptor, the application integrator. The preprocessor consists of:

- An IFC processor, which imports data, such as dimensions, materials, etc., from an IFC file into the framework. Furthermore, it can modify the IFC files, for example, create, retrieve, and delete IFC instances, and set or edit attribute values for IFC instances.
- A user interface for inputting data, which is a graphical interface for the user to input data from the IFC model or manually. Moreover, the user can access the material and weather database with this interface.

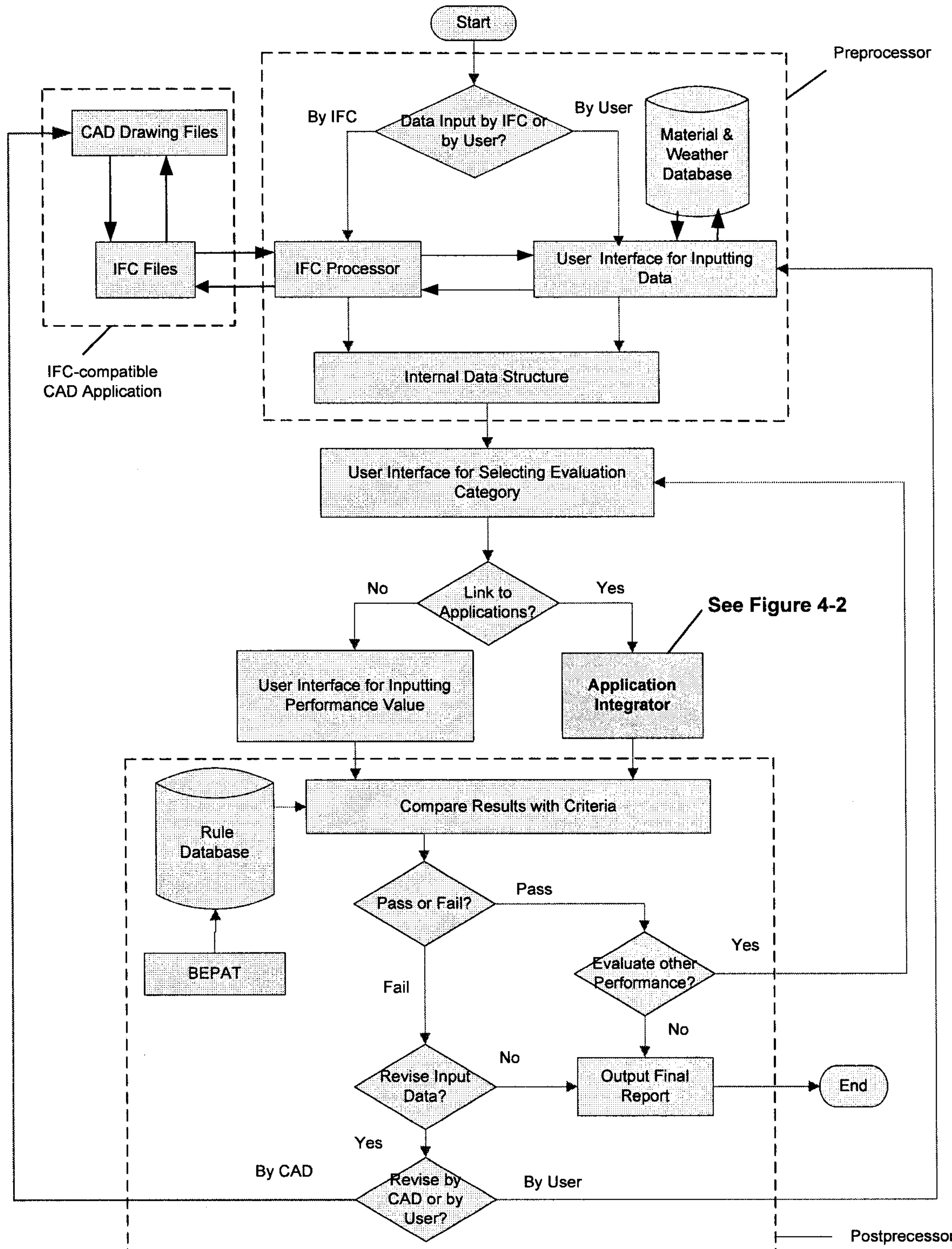


Figure 4-1. Overview of the Building Envelope Evaluation Framework

- A material and weather database, which manages the data of material and weather information in the framework.
- An internal data structure, which is used to map the data from the IFC processor and the user interface for inputting data into the memory. It could be a single class or a set of classes depending on how many applications have been linked in the application integrator. It defines all the data in the system as well as the functions that these data invoke. Moreover, it has an extensible structure which can be scaled to suit the applications that the system includes in the application integrator.

(3) A user interface for selecting evaluation category. It is a graphical interface for the user to select the categories of the evaluation performance. These categories could be one of the following: air tightness, thermal performance, moisture management performance, energy performance, structural stability of building envelope, acoustic performance, fire control of the building envelope, quality materials, quality workmanship, and maintenance. They are included in BEPAT (Horvat & Fazio, 2004).

(4) A user Interface for inputting performance value. It is a graphical interface for the user to manually input the performance value corresponding to the selected evaluation category. This interface is used when the performance value (e.g. the air leakage rate) cannot be calculated automatically by the linked simulation programs.

(5) An Application Integrator. It is used to integrate the simulation programs such as HOT2000 and MOIST3.0. It is a linkage between the preprocessor and the simulation programs as well as a receptor for the data that are transferred from the preprocessor.

Moreover, it generates performance values automatically by invoking the corresponding simulation programs. As shown in Figure 4-2, the application integrator consists of:

- A user interface for selecting the evaluation program, which is a graphic interface for users to select the simulation program corresponding to the category of evaluation performance decided by the user interface for selecting evaluation category. For instance, for assessing energy performance the user may select HOT2000; for assessing moisture management performance the user may select MOIST3.0; and for assessing air quality the user may select Airpak.
- Simulation applications, such as HOT2000, MOIST3.0, CONDENSE, EnergyPlus, WUFI-ORNL/IBP, hygIRC, Airpak, and so on.

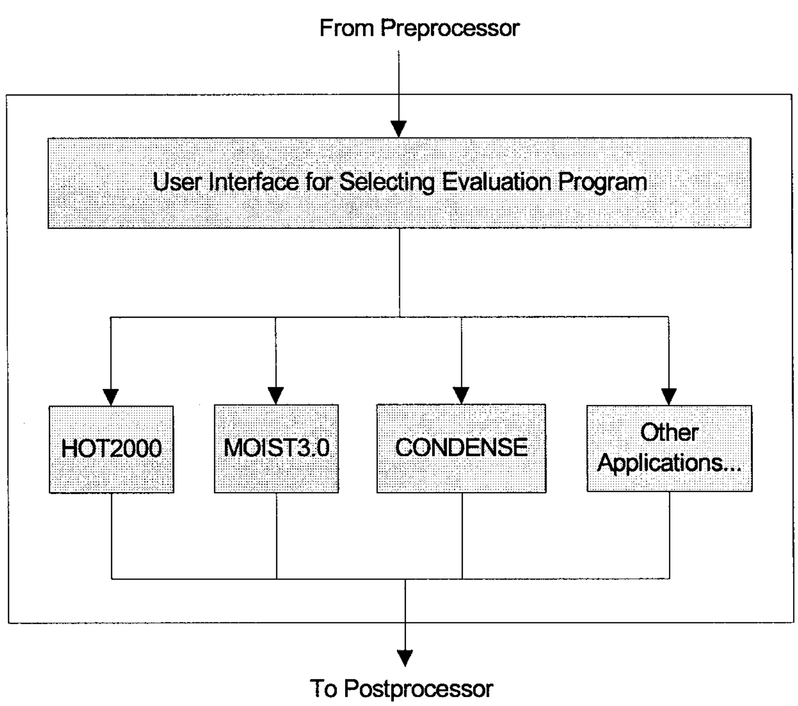


Figure 4-2. Application Integrator

(6) A postprocessor. It is a functional unit in the framework that processes the performance values of the building envelope. These performance values include two categories according to their resources. One is input by the user from user interface for inputting performance value; the other is generated automatically by simulation programs in the application integrator. The postprocessor is used to retrieve performance evaluation criteria from the rule database, compare the performance value with the retrieved criteria and create a final report to the user. The rule database contains rules extracted from the BEPAT.

4.4. The Process of Performance Evaluation

The process of building envelope performance evaluation involves inputting data into the framework, generating performance value manually or automatically, and assessing performances against evaluation criteria. The following is the process of the performance evaluation:

(1) As shown in Figure 4-1, when a user initiates the process to assess the building envelope performances, they can input the data about the building envelope by the preprocessor. These data can be input either by the IFC processor or by the user interface for inputting data and can be mapped into the memory by the internal data structure.

(2) Having input the data into the framework, the user can select the category of the evaluation performance by the user interface for selecting evaluation category. In terms of generating method, there are two types of performance values: one is input

by the users such as air leakage rate; and the other is calculated by simulation applications, such as energy consumption from HOT2000. The first type is input manually by the user interface for inputting performance value and the second one is calculated automatically by the application integrator.

(3) After generating performance values manually or automatically, the framework begins to evaluate these performance values in the postprocessor. First, it retrieves criteria from the rule database according to the category of the performance values. Then, the performance values are compared with the retrieved criteria. If the comparison passes, the user can go back to the user interface for selecting evaluation category and select another evaluation category, or output the final report directly if the user wants to stop the evaluation. On the other hand, if the comparison fails, the user can revise the input data and evaluate the performance again, or output the final report and indicate whether it fails the evaluation. There are two ways to revise the input data: one is by modifying the CAD drawing; the other is by changing the input data in the user interface for inputting data.

4.5. Data Flow with IFC

Using international information standards such as IFC is the new tendency in developing software in ACE/FM due to the need for interoperability among software applications. The concept of IFC and some relevant knowledge have been discussed in Chapter 3 in detail. This research study uses this international standard as its data model. The data flow in the framework is show in Figure 4-3. Its process is described as the following:

(1) The data of a building is stored in CAD drawing files. It could be in 2D or 3D model in .DWG or .DXF format.

(2) With the IFC-compatible-CAD applications such as Architectural Desktop and ArchiCAD, data can be transferred into IFC model, physically in ISO 10303-21 clear text format (ISO, 2002).

(3) The data in the IFC model can be extracted into an internal data structure in which data in the IFC model can be mapped into the memory by the IFC toolbox.

(4) The data in the memory are written into input files such as *.dat files for MOIST3.0 and in *.V70 and *.V80 files for HOT2000.

(5) The output data are generated by invoking the applications.

(6) Finally, these output data are compared with the criteria, and the evaluation summary output is generated. If the evaluation fails, the user can revise the input data from the CAD drawing files or IFC files.

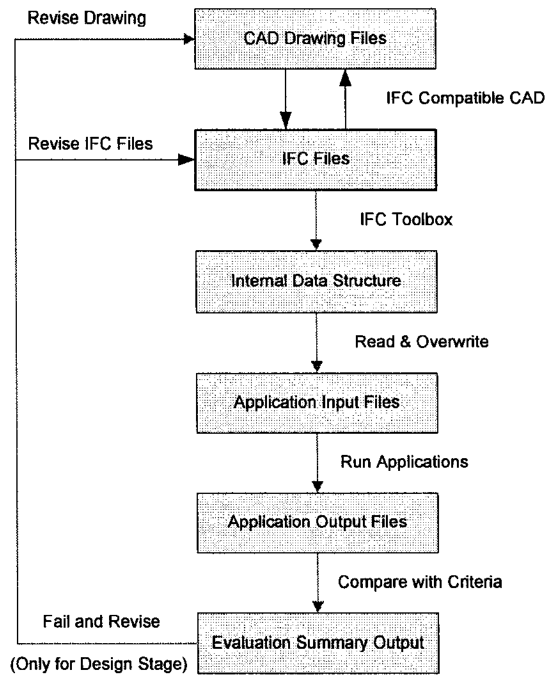


Figure 4 -3. Data Flow with IFC Model in the Evaluation Framework

4.6. Evaluation Criteria

The results from the evaluation applications need a set of criteria to evaluate. As introduced in Chapter 2, several protocols already exist to evaluate the overall or part of the performance of a house, for example, the P-mark from Sweden (SP, 2005), the Housing Quality Assurance Law (HQAL) from Japan (Eastin et al., 2000), and the European Technical Approval Guidelines ETAG 007 from Europe (EOTA, 2002). In Canada, there is the R-2000 program (NRCan, 2005) and the Novoclimat from Quebec (Quebec Agency for Energy Efficiency, 2005). To advance the concept of overall building performance, Fazio's research group at Concordia University first undertook the classification of the requirements and standards governing the building performance resulting in the BEPAT (Horvat & Fazio, 2004). Its purpose is to evaluate the overall performance for light-frame site-built or prefabricated residential and small commercial building envelopes. The following functional requirements are included in this protocol: structural stability, air tightness, moisture management performance, thermal performance, energy performance, acoustic performance and fire control of the building envelope. At present, the criteria are based on Montreal's climatic, technical and social environment. However, it is envisioned to represent a framework for developing similar protocols and assessment tools for examining performance of building envelopes under different parameters, priorities, technologies and building traditions that exist in various regions and countries in the world (Horvat & Fazio, 2004). The proposed framework applies the BEPAT as its evaluation criteria.

Chapter 5

Implementation of the Framework

5.1. Introduction

In order to prove the concept of the evaluation framework presented in Chapter 4, a prototype system is developed (He et al., 2005; Hammad et al., 2005). This chapter describes the development of this prototype system. At this stage, only part of the framework presented in Chapter 4 has been implemented. Figure 5-1 shows the implemented prototype system. In contrast with the full framework shown in Figure 4-1, the following simplifications have been made in Figure 5-1:

(1) The IFC processor in Figure 5-1 now can import only data from the IFC files and cannot modify the IFC files by the general data input (user interface for inputting data). Moreover, these imported data are limited only to geometry and material layer data. Other data can be input manually by the user.

(2) The system can evaluate only the performance values which are generated automatically by the application integrator. In other words, the user interface for selecting evaluation categories and the user interface for inputting performance values in Figure 4-1 are not implemented.

(3) The application integrator can link only to two simulation programs: HOT2000 and MOIST3.0.

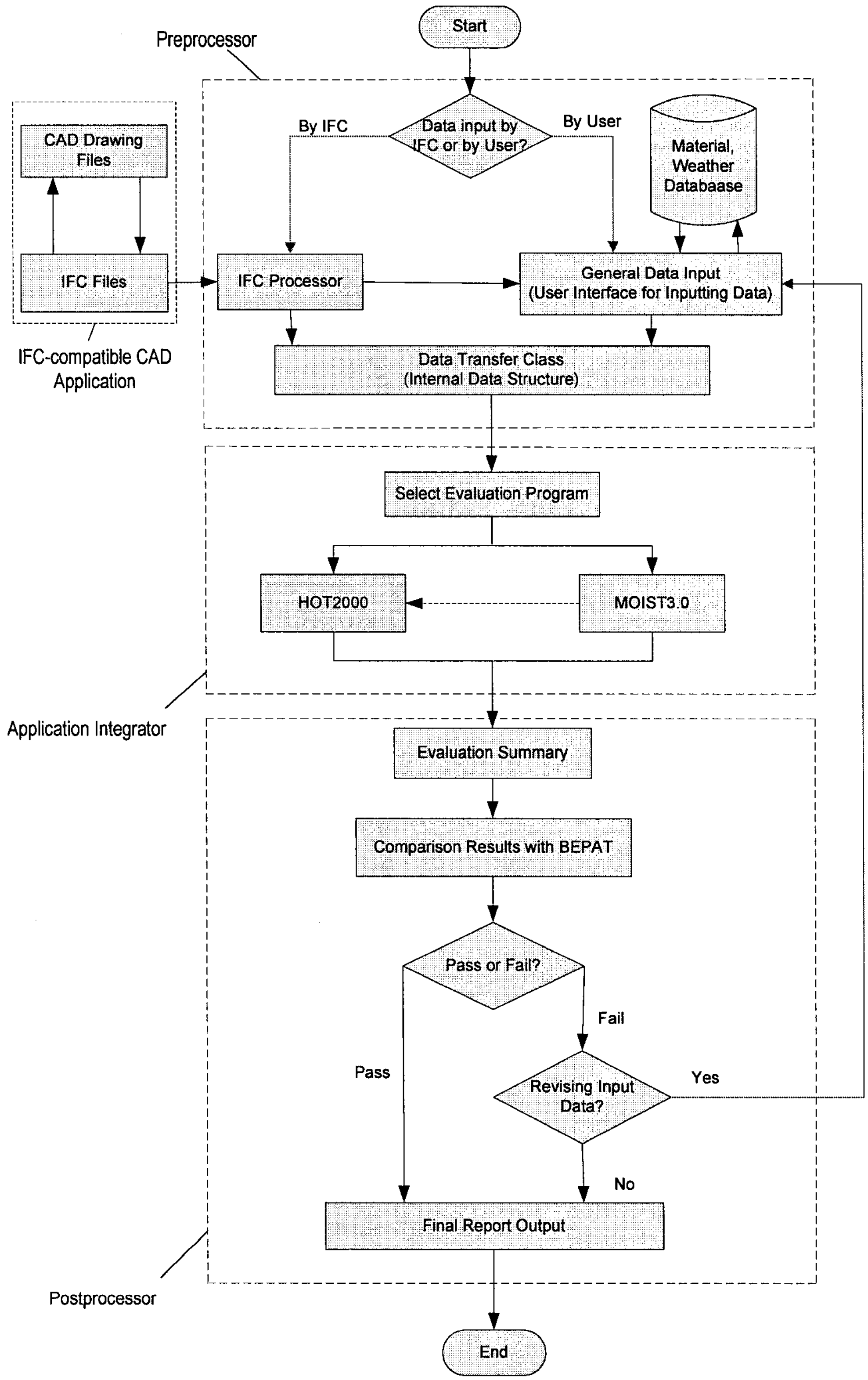


Figure 5-1. Implemented Prototype System

(4) The prototype system simplifies some complicated input data, such as HVAC systems of a house, by defining them using default values.

(5) The comparison in the postprocessor is limited to one part of BEPAT by extracting the rules (clauses) from the system. Knowledge-based code compliance checking is not included in the prototype system at this moment.

The following sections discuss the implemented prototype system in detail based on these simplifications.

5. 2. Software Engineering Design

The implemented prototype system is a Windows application and is developed under the Microsoft • NET 2003 platform, which has a user-friendly Integrated Development Environment (IDE) and is very easy to use by software developers. The Microsoft • NET is described as a next generation Windows services project because it embraces Microsoft's core strategy, plans, and vision for the foreseeable future. The core of this platform is the Microsoft • NET framework. It consists of three components:

(1) Microsoft Intermediate Language (MSIL). All codes in one or more languages in a project will be compiled into a more abstract, trimmed-down form before it is executed. This means that the Microsoft • NET supports multi-language in a project.

(2) Common Language Runtime (CLR). It executes the MSIL and deal with all jobs that Windows and IIS (Internet Information Services) require. It uses Just-In-Time (JIT) compiler to compile to true machine code. Currently, CLR is only compatible with

Windows (9x, NT, 2000, XP), but its new version is being built for other operating systems.

(3) Microsoft • NET Framework Class Library. It contains an enormous amount of codes that the users can include in any of their programs to simplify all sorts of useful tasks (Goode et al., 2002).

As for the database in the framework, Microsoft Access is chosen as the Database Management System (DBMS) due to its simplicity. In the future, if more simulation applications will be integrated and more input data will be needed, other more powerful DBMS could be introduced into the framework.

5.3. Data Transfer from CAD Drawings to IFC Files

Data transfer from CAD drawings to IFC files could be achieved by IFC-compatible CAD applications. As the IFC has been determined as the information model in the proposed research study, this raises another question: how to transfer the data of a building in CAD drawings into an IFC data model? As described in Chapter 3, the IFC is physically implemented in a clear text file format, which complies with the regulations prescribed in STEP ISO-10303-21 (ISO, 2002). Since there is a huge amount of information related to a building, the data of a building are usually expressed in thousands of lines in an IFC text file. As a result, it is almost impossible to generate an IFC text file from CAD drawings manually. Instead, this task is normally implemented by computer software automatically. The Autodesk's Architectural Desktop with its IFC2x Utility (Inopso GmbH, 2005) and Graphisoft's ArchiCAD with its add-on interface (Graphisoft, 2005) are two typical applications that can generate an IFC file from the CAD drawings.

5.3.1. Autodesk's Architectural Desktop with its IFC2x Utility

The Autodesk's Architectural Desktop with its IFC2x Utility is one application that automatically generates an IFC file from CAD drawings. It is developed by Autodesk and Inopso GmbH. They cooperated to develop the IFC2x Utility, a CAD-IFC-translator, which allows the building industry professionals to freely share intelligent data. These data could be created in Autodesk Architectural Desktop 3.3, 2004, and 2005 software as well as in other software developed by other IAI members. Accordingly, with the IFC2x Utility, Autodesk can support industry standards for integration and interoperability (Autodesk, 2005).

5.3.2. Graphisoft's ArchiCAD with its Add-on Tool for IFC

Graphisoft's ArchiCAD with its add-on tool for IFC is another application to automatically generate an IFC file from the CAD drawings. It is developed by the Graphisoft, which has been a member of the IAI since 1996. With its add-on interface, ArchiCAD provides the users with a tool which is compatible with the releases IFC 1.5.1 and IFC 2.0. Moreover, the Graphisoft is currently implementing the support of the release IFC2x and IFC2x2, and the beta version of the add-on tool for these IFC releases can be downloaded freely from the Web (Graphisoft, 2005).

5.4. Implementation of the Preprocessor

As presented in Chapter 4, the *preprocessor* is a functional unit in the framework that processes the input data for the building. It is used to access an IFC file generated from the CAD application; input data from end-users; store data to, and retrieve data from, the database; and build a middle layer (*data transfer class*) to transfer data to its receptor, the *application integrator*. As shown in Figure 5-1, the main components in the preprocessor are: *IFC processor*, *general data input*, and *data transfer class*.

5.4.1. Importing Data from an IFC File into the Framework

After an IFC-compatible CAD tool such as Architectural Desktop or ArchiCAD has generated an IFC file, the system should import the data from the IFC model. As mentioned in Chapter 3, accessing the IFC file needs the help of the IFC toolbox, which is an interface or middleware to read from and write to the IFC model. In this project, the Active Toolbox from the Eurostep Group is used as a middleware to access the IFC model (Eurostep Group, 2005). The introduction about this toolbox is presented in Appendix A. So far, only geometry data, such as the number of stories, the total length and average height of the exterior walls, and material layer data, such as material name and layer thickness, are imported into the system.

__The Process of Importing Data from an IFC Model__

The process of importing data from the IFC model involves searching for an IFC file by the user and extracting data from the selected IFC files. It includes the following steps:

(1) The user runs the system and clicks the button, *start the framework*. Then, a small Window pops into the center of the screen, as shown in Figure 5-2. It asks the user whether to import data from an IFC file. If the user answers "No", a new graphical interface, *general data input*, is started in which the user can input data manually.

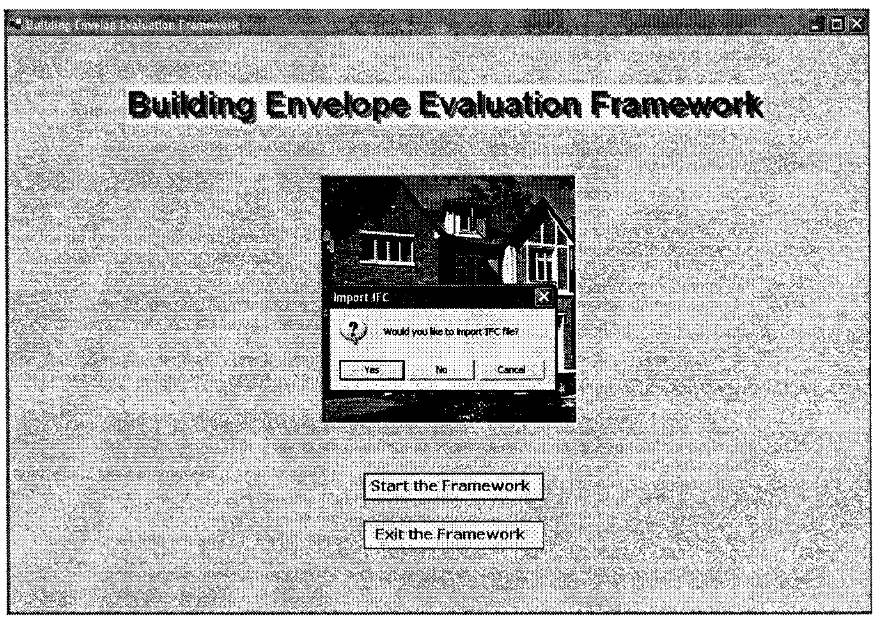


Figure 5-2. The Interface of Starting the Import of an IFC File

(2) If the user answers "Yes", another new graphical interface, *import IFC text file*, in which the user can import geometry and material layer data from the IFC model, pops into the center of the screen, as shown in Figure 5-3. There are two functionalities for this interface: one is searching for an IFC file, and the other is firing an event to import data from an IFC file.

(3) By clicking the button, *browse*, a Window, *search IFC file*, as shown in Figure 5-4, will display on the screen. It helps the user to find and select an IFC file in the system.

(4) By clicking the button, *import IFC*, the geometry and material layer data in the IFC file, which is selected in step 3, are imported into the system. Moreover, they are mapped into the memory by the *data transfer class*. These imported data are also displayed in the textbox, as shown in Figure 5-3.

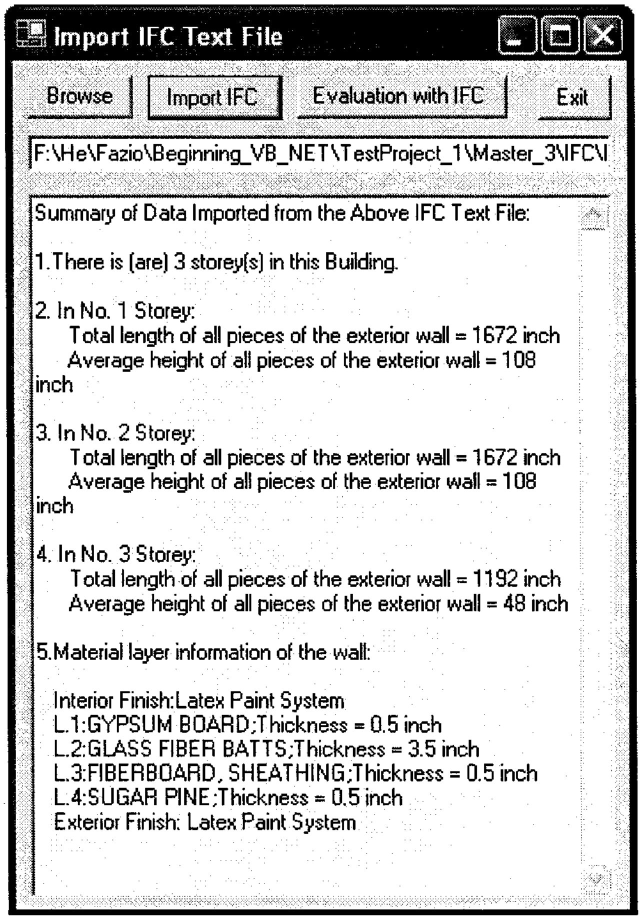


Figure 5-3. The Interface of Importing an IFC File

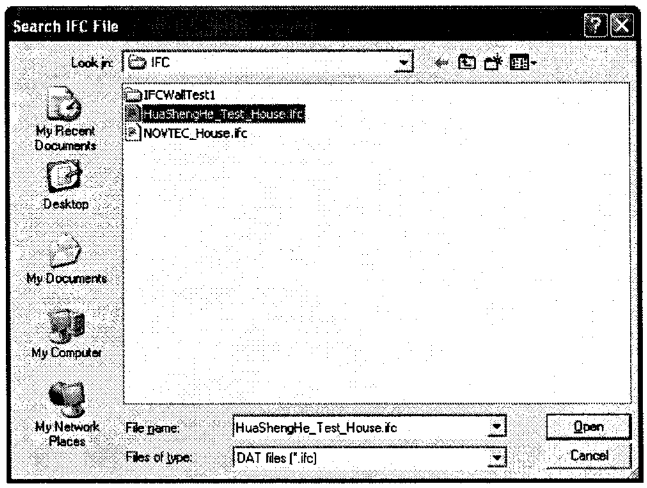


Figure 5-4. The Interface of Searching for an IFC File

The data display in Figure 5-3 is about a sample building. The textbox shows the number of stories, the total length and average height of the exterior walls in each floor, the exterior and interior finishes, the material names and layer thicknesses of the exterior walls.

The Algorithm of Extracting the Height and Length of the Exterior Walls from an IFC File

In order to set up the algorithm of extracting the height and length of the exterior wall from an IFC file, it is necessary to analyze the IFC entities related to the height and length of the wall. Figure 5-5 shows a fragment of an IFC text file in which all entities are related to the wall. The format of each line can be found in section 3-7. The definition and attribute list of each entity are shown in Table 5-1. The flowchart in Figure 5-6 explains the process of extracting the data. It includes the following steps:

(1) Getting the number of stories in the building.

(2) Getting the IfcBuildingStorey (e.g. #34) for the first storey, which the Elevation is equal to zero.

(3) Getting the IfcRelContainedInSpatialStructure (e.g. #6543) for the first storey, which the IfcBuildingStorey is equal to that in Step 2 (e.g. #34).

(4) Getting all RelatedElements (e.g. #61, #108, …, #6054, #6109) in IfcRelContainedInSpatialStructure (e.g. #6543).

(5) Getting all IfcWallStandardcase (e.g. #61) in all RelatedElements and then filtering them with Exterior Wall.

(6) Getting the IfcBoundingBox (e.g. #53) for each IfcWallStandardcase(e. g. #61).

(7) Getting the height and length for each IfcBoundingBox.

(8) Going back to Step (2) and repeating the same process to get the second or other storey for the building.

(9) Outputting the results.

```
…
#34 = IFCBUILDINGSTOREY ('1SaXse6wr0IeKTgRAJX_1T', #6, '', $, $, #33, $, $,
.ELEMENT., 0.);
…
#53 = IFCBOUNDINGBOX (#52, 6175.000000000001, 150.0000000000003, 4000.);
…
#56 = IFCSHAPEREPRESENTATION (#55, '', 'BoundingBox', (#53));
#57 = IFCPRODUCTDEFINITIONSHAPE ($, $, (#42, #51, #56));
…
#61 = IFCWALLSTANDARDCASE ('1VU1y_6tDBVRCjdpZenjMQ', #6, 'Sein\X\E4 001',
'Exterior Wall' , $, #60, #57, $);
…
#6543 = IFCRELCONTAINEDINSPATIALSTRUCTURE ('3tXycwPR5B9hv0PUvi1vUR', #6,
'BuildingStoreyContainer', 'BuildingStoreyContainer for Building Elements', (#61, #108, #164,
#219, #267, #286, #471, #535, #582, #637, #693, #756, #804, #860, #916, #979, #999, #1165,
#1331, #1540, #1560, #1726, #1896, #2062, #2228, #2432, #2452, #2618, #2784, #2950, #3116,
#3315, #3334, #3491, #3694, #3713, #3869, #4034, #4199, #4397, #4416, #4618, #4638, #4795,
#4961, #5127, #5334, #5353, #5509, #5691, #5734, #5776, #5818, #5860, #5902, #5944, #5986,
#6054, #6109), #34);
…
```

Figure 5-5. A Fragment of an IFC File Related to the Wall

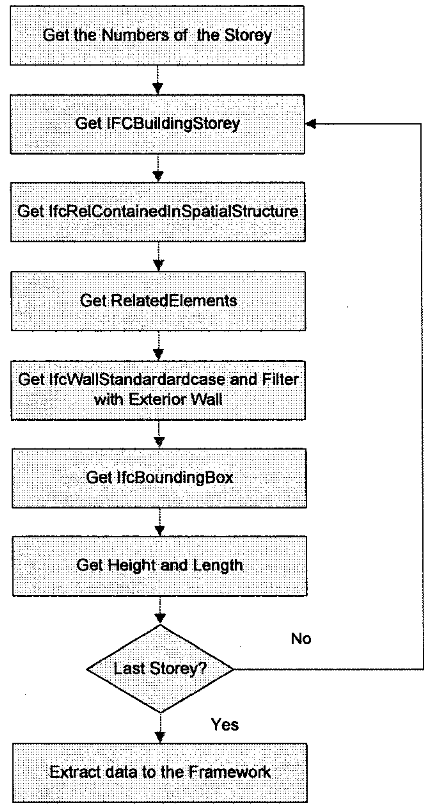


Figure 5-6. The Flowchart of Extracting the Height and Length of the Exterior Walls from an IFC File

The Algorithm of Extracting the Material Layers of the Exterior Walls from an IFC File

Like extracting the height and length of the exterior walls, it is necessary to analyze the IFC entities related to the material of the wall before setting up the algorithm of extracting the material layers from an IFC file. Figure 5-7 shows a fragment of an IFC text file in which all entities are related to the material. The format of each line and the entity definition related to the material can be found in section 3.7. The flowchart in Figure 5-8 explains the process of extracting the data. It includes the following steps:

(1) Getting all IfcWallStandardcase(s) (e.g. #107) which have attributes "Exterior Wall" (this procedure is the same as that in extracting the height and length of the exterior wall).

(2) Getting the IfcRelAssociatesMaterial (e.g. #128) related to the wall in Step (1).

(3) Getting the IfcMaterialLayerSetUsage (e.g. #127).

(4) Getting the IfcMaterialLayerSet (e.g. #126).

(5) Getting the IfcMaterialLayer (e.g. #125) and reading the thickness of the layer.

(6) Getting the IfcMaterial (e.g. #113) and reading the material name.

(7) Going back to IfcMaterialLayerSet (e.g. #126) and dealing with another material layer.

(8) Outputting the results.

```
...
#107=IFCWALLSTANDARDCASE('14euheRDfFQxwQGdktUE5y',#19,'','Exterior Wall','',#72,#81,$);
#111=IFCMATERIAL('Latex Paint System');
#112=IFCMATERIALLAYER(#111,0.,.F.);
#113=IFCMATERIAL('GYPSUM BOARD');
#114=IFCMATERIALLAYER(#113,0.5,.F.);
#115=IFCMATERIAL('GLASS FIBER BATTS');
#116=IFCMATERIALLAYER(#115,4.,.F.);
#117=IFCMATERIAL('POLYSTYRENE, EXTRUDED');
#118=IFCMATERIALLAYER(#117,1.5,.F.);
#119=IFCMATERIAL('PLYWOOD, EXTERIOR GRADE');
#120=IFCMATERIALLAYER(#119,0.5,.F.);
#121=IFCMATERIALLAYER(#117,1.5,.F.);
#122=IFCMATERIAL('BRICK');
#123=IFCMATERIALLAYER(#122,0.5,.F.);
#124=IFCMATERIAL('No Paint No WallPaper');
#125=IFCMATERIALLAYER(#124,0.,.F.);
#126=IFCMATERIALLAYERSET((#112,#114,#116,#118,#120,#121,#123,#125),'Standard');
#127=IFCMATERIALLAYERSETUSAGE(#126,.AXIS2.,.POSITIVE.,0.);
#128=IFCRELASSOCIATESMATERIAL('1wftdzun5EO9H7W1nOP20L',#19,$,$,(#107),#127);
...
```

Figure 5-7. The Fragment of an IFC File Related to the Material

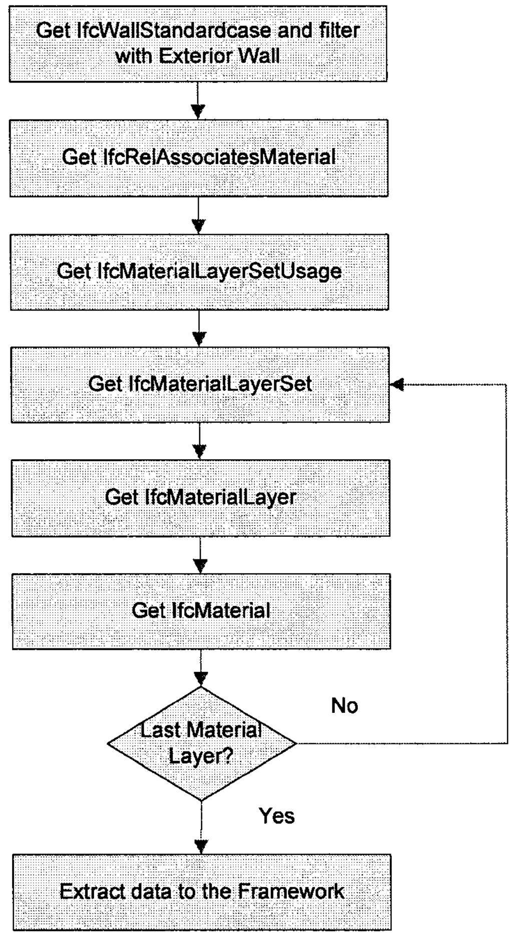


Figure 5-8. The Flowchart of Extracting the Material Layer of the Exterior Wall from an IFC File

Table 5-1. IFC Entities Related to Walls (From IAI, 2005a)

Entity	Definition	Attribute List (The names of the attribute are listed in order)
IfcBuildingStorey	The building storey has an elevation and typically represents a (nearly) horizontal aggregation of spaces that are vertically bound.	(GlobalId, OwnerHistory, Name, Description, ObjectType, ObjectPlacement, Representation, LongName, CompositionType, Elevation)
IfcBoundingBox	Every semantic object having a physical extent might have a minimum default representation of a bounding box. The bounding box is therefore also used as minimal geometric representation for any geometrically represented object. Therefore the *IfcBoundingBox* is subtyped from *IfcGeometricRepresentationItem*.	(Corner,Xdim,Ydim,Zdim)
IfcProductDefinitionShape	A product definition shape identifies a product's shape as the conceptual idea of the form of a product.	(Name, Description, Representations)
IfcShapeRepresentation	The *IfcShapeRepresentation* represents the concept of a particular geometric representation of a product or a product component within a specific geometric representation context.	(ContextOfItems, RepresentationIdentifier, RepresentationType, Items)
IfcWallStandardCase	The standard wall (*IfcWallStandardCase*) defines a wall with certain constraints for the provision of parameter and with certain constraints for the geometric representation. The *IfcWallStandardCase* handles all cases of walls, that have a single thickness along the path, i.e.:parallel sides for straight walls and co-centric sides for curved walls.	(GlobalId, OwnerHistory, Name, Description, ObjectType, ObjectPlacement, Representation, Tag)
IfcRelContainedInspatialStructure	This objectified relationship, *IfcRelContainedInSpatialStructure*, is used to assign elements to a certain level of the spatial project structure. Any element can only be assigned once to a certain level of the spatial structure. The question, which level is relevant for which type of element can only be answered within the context of a particular project and might vary within the various regions.	(GlobalId, OwnerHistory, Name, Description, RelatedElements, RelatingStructure)

5.4.2. General Data Input

General data input is the graphical interface to input data manually for the simulation programs. Some tab pages such as *geometry data* and *layer information* also have functionalities of importing data from an IFC model.

By the assistance of the IFC toolbox, the system can get the data from an IFC file. This process enables the system to transfer data from CAD drawing to the *preprocessor* seamlessly. However, data extracted by this way do not meet all the requirement of the evaluation program, for instance, in the case when the CAD drawings do not include the data (e.g., HVAC system) which the simulation programs need, or the data needed in the programs are not included in the current IFC version, inputting data manually is necessary. A *general data input* is such an interface that is designed for the system to obtain data not only by importing from an IFC file but also by inputting them manually.

The *general data input* includes five tab pages: *location and general*, *material database*, *geometry data*, *layer information*, and *save and start*.

(1) *Location and general*. This tab page is used mainly to input weather information for the evaluated building envelope. As show in Figure 5-9, the input data include: weather locations, envelope location, orientation, unit, and indoor temperature and RH (Relative Humidity).

- Weather locations. They are three comboboxes that list the weather information invoked in MOIST3.0 and HOT2000. They include WYEC (Weather Year for Energy Calculation) in MOIST3.0, and weather region and weather city in HOT2000.

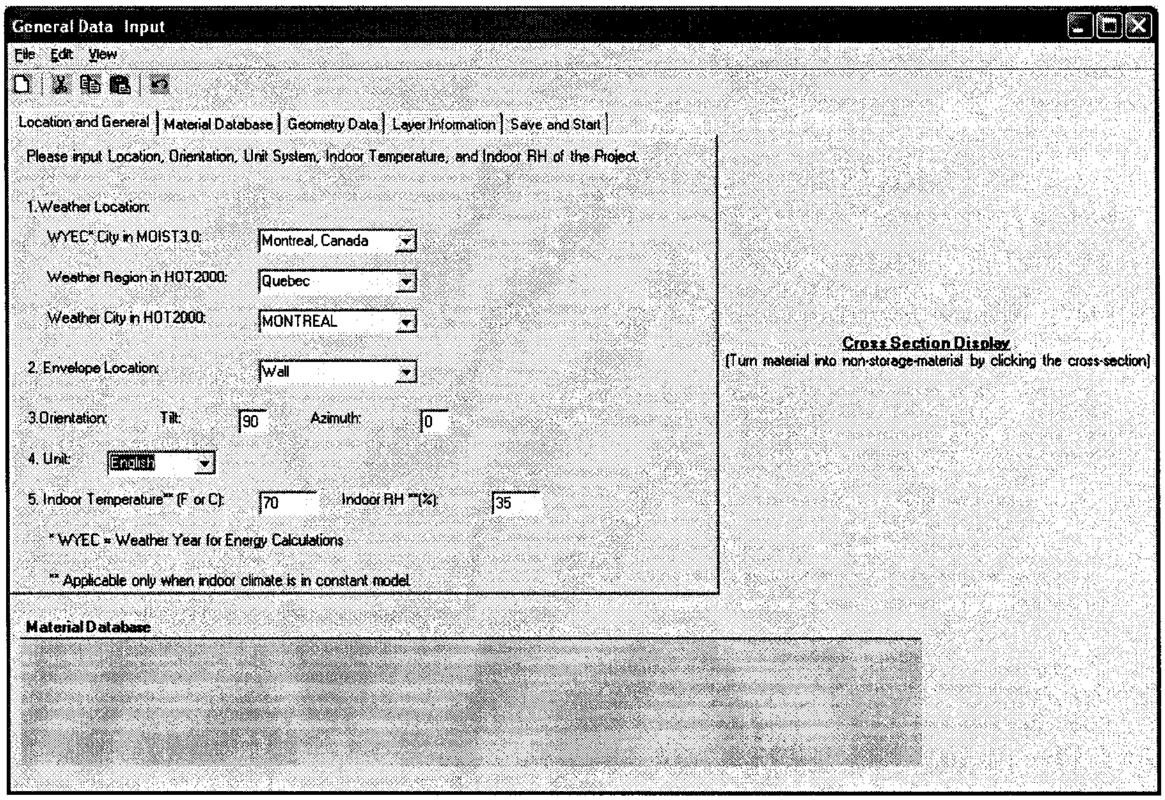


Figure 5-9. Tab Page *Location and General* in General Data Input

- The WYECs are the hourly weather data from ASHRAE. They are weather data for 51 cities in the United States and Canada (e.g., Vancouver, Edmonton, Winnipeg, Toronto, and Montreal in Canada).
- The weather regions in HOT2000 are the provinces or territories in Canada (e.g., Quebec, Ontario, Manitoba, British Columbia, etc.).

- The weather cities in HOT2000 are the main cities or areas in the province or territory in Canada (e.g., Kuujjuaq, Bagotville, Montreal, Poste De La Baleine, Quebec, Schefferville, Sept lles, Sherbrooke, and Val D'or in Quebec).

- Envelope location. It is a combobox to select the type of the evaluation assembly with MOIST3.0. It has only two options: wall and roof (At this moment, only wall has been implemented. Moreover, the prototype system is limited to evaluate a house with only one type of wall.).
- Orientation. It includes two textboxes, tilt, and azimuth which are the angles to determine the orientation of the evaluated assembly (wall or roof).
- Unit. It is a combobox to define the unit system. It includes IS and English.
- Indoor temperature and Relative Humidity (RH). They are two textboxes to input the indoor temperature and relative humidity. They are applicable only when indoor mode is in constant mode but are not applicable when the indoor mode is in variable mode.

(2) *Material database.* Many materials are used in building envelopes. Moreover, new materials may be invented and applied to building envelopes any time. Accordingly, it is almost impossible to design a database which can accommodate all materials that are needed in building envelopes. This proposed framework has designed a mechanism to enable the users to update the material database conveniently. The tab page *material database* is just such a graphical interface to manage the material

database in the framework. As show in Figure 5-10, it has four options to access the material database:

- Option 1. *Display all material in the database.* By clicking this button, all materials in the database are listed with a table in the bottom of the tap page.
- Option 2. *Insert material.* By clicking this button, a new material can be added into the database.
- Option 3. *Modify material.* By clicking this button, the selected material can be modified.
- Option 4. *Delete material.* By clicking this button, the selected material can be deleted.

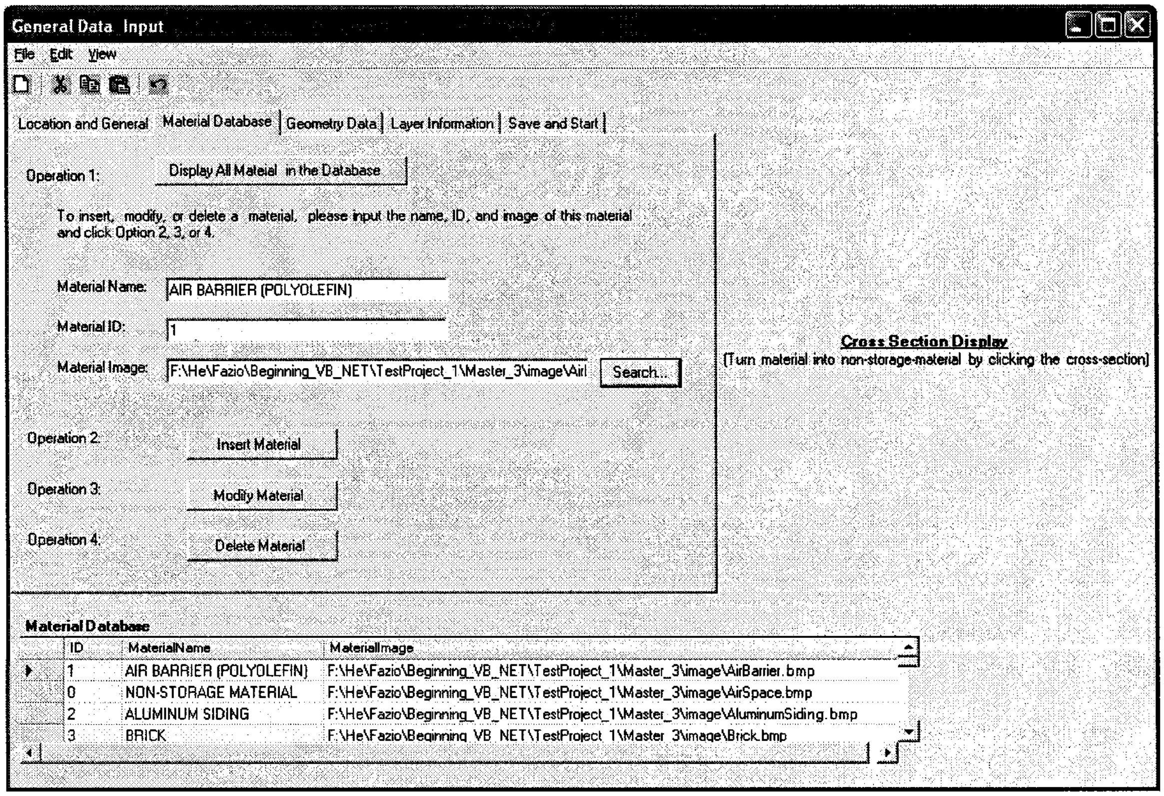


Figure 5-10. Tab Page *Material Database* in General Data Input

In order to insert, modify, or delete a material, it is necessary to input data for the material, such as the name, ID, and the path of the image file. These data can be input by three textboxes in the middle of the tab page. The Name and the ID can be typed directly in the textboxes, and the path of the image file can be searched by clicking the button, *searching*, which is beside the material image textbox. Once the user has clicked this button, a searching image window will pop up, as shown in Figure 5-11. Then, the user can browse the image files in the system and select an image to display the new material.

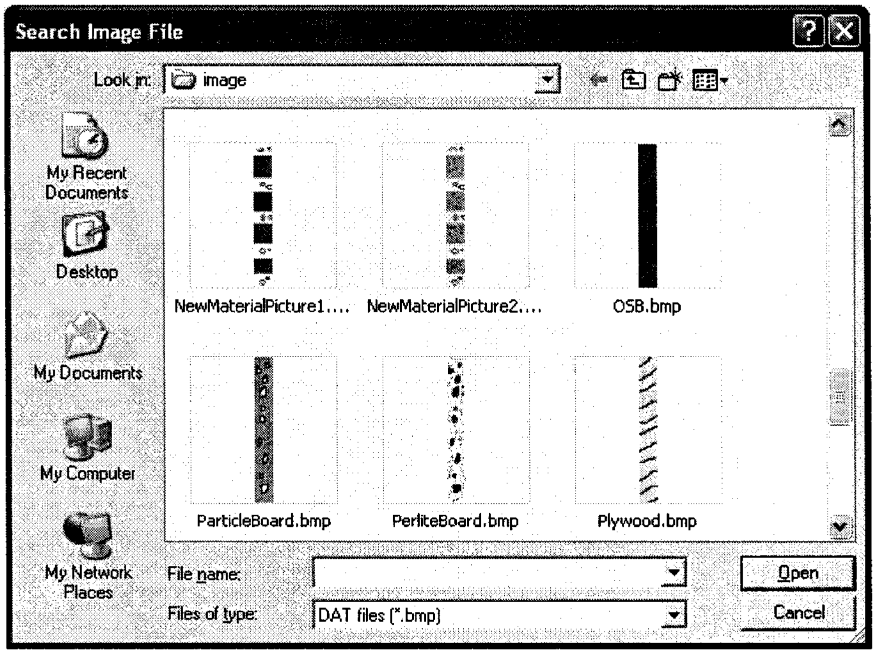


Figure 5-11. Window for Searching an Image File for a New Material

(3) *Geometry data.* This tab page is used to input the geometry data such as the number of the stories, the total length and average height of the exterior wall in each floor. As shown in Figure 5-12, the user can input these data manually. He/she can also get these data from the IFC model by clicking the button, *get geometry data from IFC.*

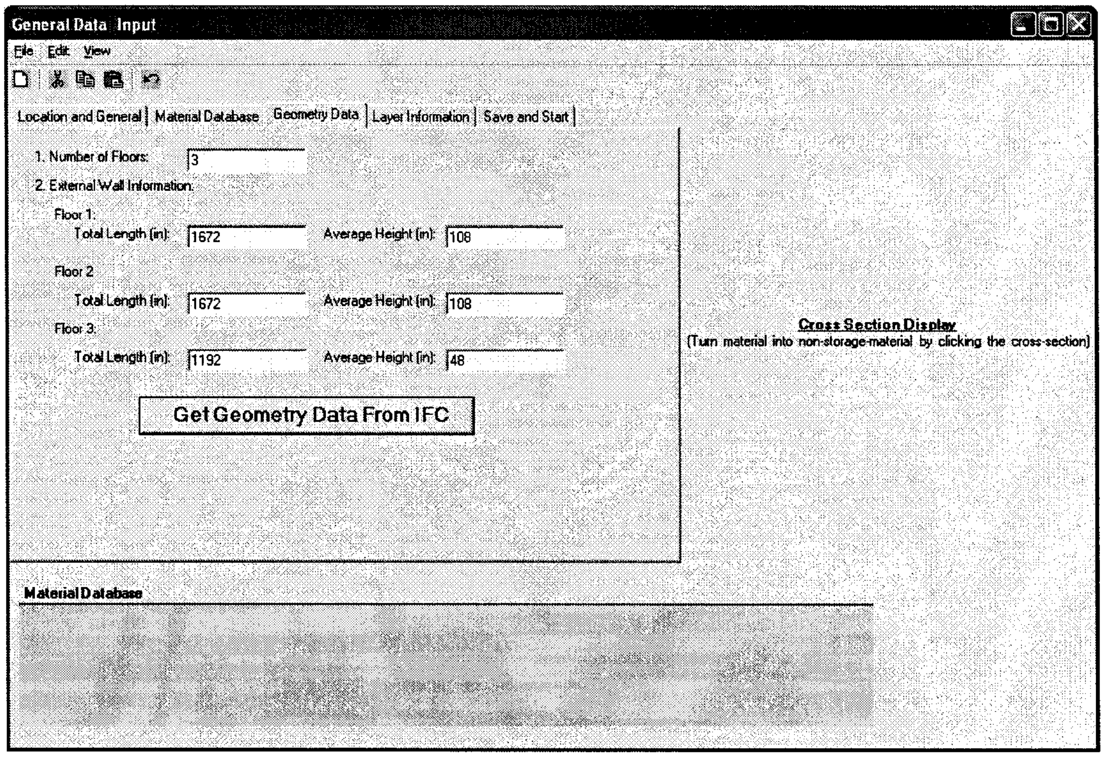


Figure 5-12. Tab Page *Geometry Data* in General Data Input

(4) *Layer information.* This tab page is used to input the layer information of the evaluated assembly (Wall or Roof). So far, the system can handle up to six layers plus two finishes (interior and exterior). As show in Figures 5-13 and 5-14, the information of the layers and finishes is listed from interior to exterior. The use can

input these data manually. He/she can also get these data from the IFC model by clicking the button, *get layers from IFC.*

- The interior and exterior finish. They are two comboboxes which list the category of the finish which includes:
    - Latex Paint System
    - Oil-base Paints
    - Vapor Retarder Paints
    - Vinyl Wallpaper
    - No Paint No Wallpaper
- Layer 1 to Layer 6. They are six comboboxes which list all material names in the database. The default material name list is the same as that in MOIST3.0, as shown in Appendix B.
- Thickness: a textbox to input the thickness of the layer.
- MC: a textbox to input the initial moisture content of each layer.
- Wood-base layer: a combobox to identify whether the layer is wood-based material.
- Thermal resistance: a textbox to input the thermal resistance of the non-storage material. It is only applicable to non-storage material (Figure 14).
- Permeance: a textbox to input the permeance of the non-storage material. It is also only applicable to non-storage material (Figure 14).

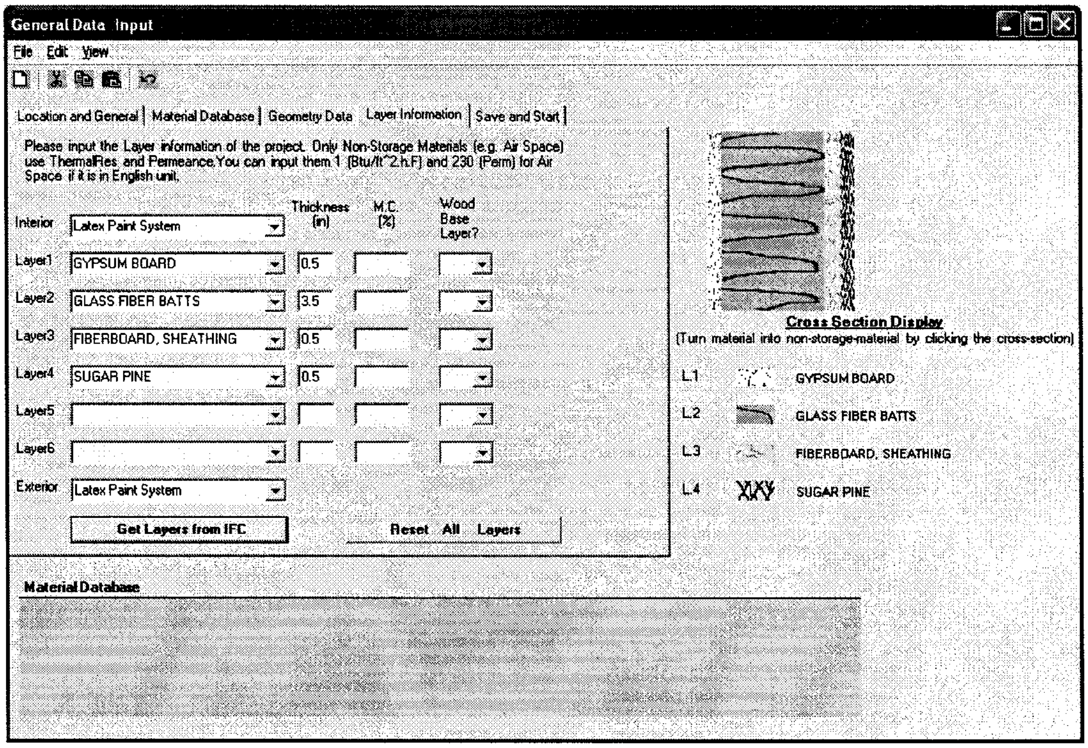


Figure 5-13. Tab Page *Layer Information* in General Data Input

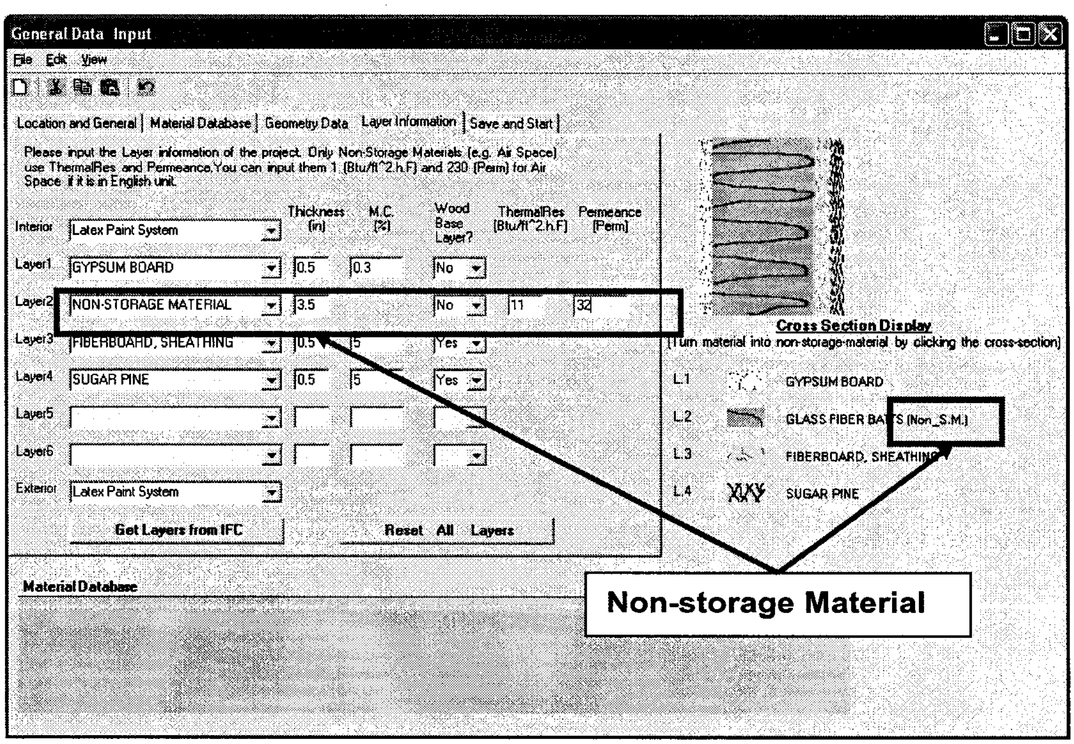


Figure 5-14. Tab Page *Layer Information* with Non-storage Material

Furthermore, an image to express the cross-section of the wall or roof is displayed in the right hand side of the tab page. If the user wants to change the material into non-storage material, he/she can do this by clicking the material in the cross-section. In addition, if the user wants to change all the material layers, he/she just clicks the button, *reset all layers*, and all the material information and cross section will be cleared in the interface.

Here, the non-storage material is the insulation material with low density and cannot store heat and moisture (e.g. glass fiber blanket insulation and cellulose insulation). The difference between Figure 5-13 and 5-14 is that, in Figure 5-14, the second layer, glass fiber batts, has been changed into non-storage material by clicking its image in the cross-section. Moreover, in Figure 5-14, the textbox of MC for layer 2 has disappeared. However, two new textboxes, *thermal resistance* and *permeance*, are added in the interface. The purpose to change the insulation of layer 2 into non-storage material is to get convergence in the calculation of MOIST3.0.

(5) *Save and start.* As shown in Figure 5-15, this tab page has the following four functionalities:

- Drawing the cross-section proportionally. This functionality is applicable when the material layer data are input manually, and is not applicable when the material layer data are imported from IFC.
- Saving the data input from the *general data input* into the internal data structure, *data transfer class*.

- Starting the evaluation after the data input has finished.
- Exiting the data input model if needed.

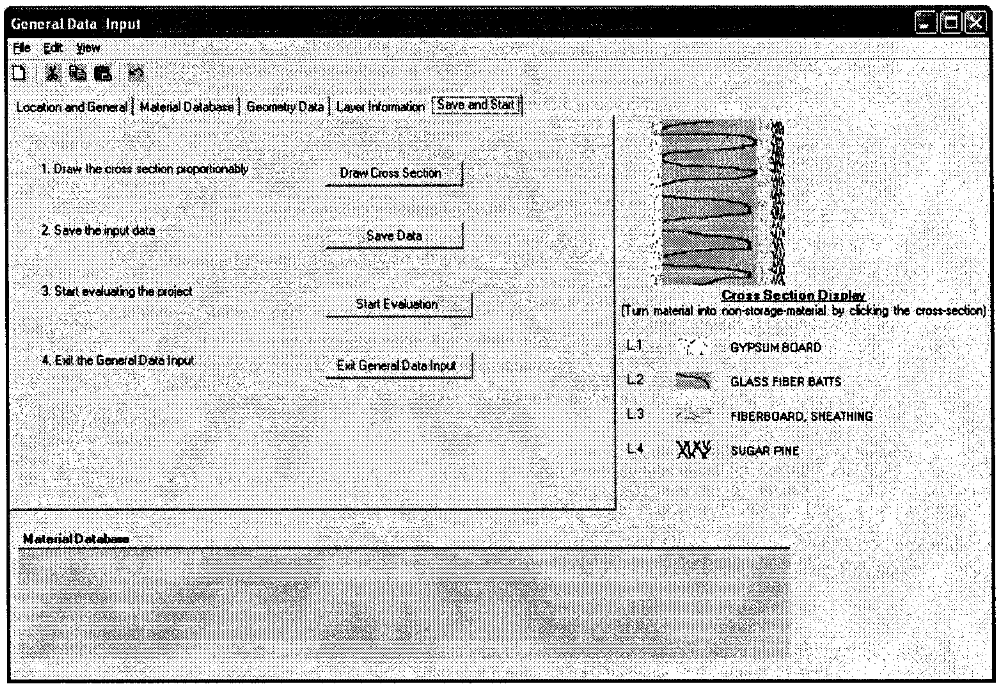


Figure 5-15. Tab Page *Save and Start* in General Data Input

5.4.3. Data Transfer Class

Data transfer class is the internal data structure to mange the data in the memory. As introduced in the above sections, there are two ways to input data into the system: one is importing from IFC and the other is inputting manually. In order to invoke these data by the simulation programs, it is necessary to create a data structure to map these data into the memory. *Data transfer class* is such a class or a set of classes that manage the data extracted from the IFC or input manually from the *general data input*. The number of the class depends on how many applications have been linked in the *application integrator*. It defines all the data that the system could use as well as the functions that these data could invoke. Moreover, it has an extensible structure which can be scaled to suit the applications that the framework includes in the *application integrator*. Figure 5-16 shows the function of the *data transfer class*.

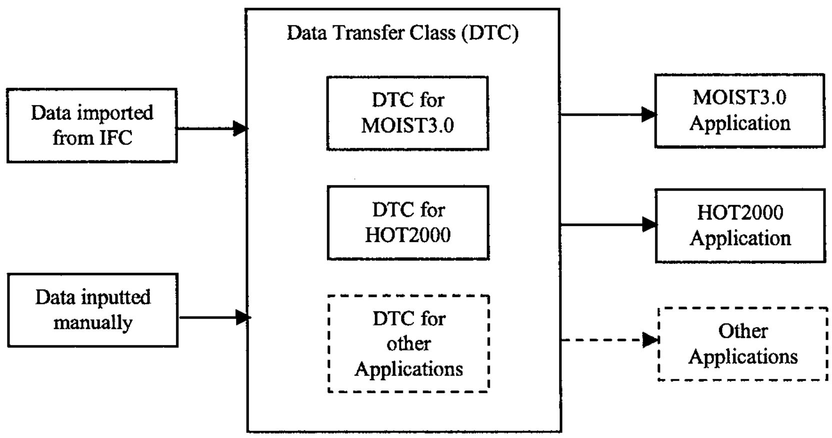


Figure 5-16. ***Data Transfer Class***

5.5. Implementation of the Application Integrator

The implementation of the *application integrator* involves the linkage of the simulation programs, HOT2000 and MOIST3.0. As presented in Chapter 4, the *application integrator* integrates the simulation programs and is a linkage between the *preprocessor* and the simulation programs as well as a receptor for the data that are transferred from the *preprocessor*. Nonetheless, linking to the simulation applications needs technical support from their developers by providing APIs (Application Programming Interfaces) so that the framework could access the applications. Currently, the APIs for MOIST3.0 and HOT2000 (Batch version) are available.

5.5.1. Selecting Simulation Programs

As shown in Figure 5-17, *electing simulation programs* is the graphical interface for controlling the evaluation process with simulation programs. It has four control buttons:

(1) *MOIST3.0 evaluation* which is used to evaluate the assemblies (e.g. wall) with MOIST3.0.

(2) *HOT2000 evaluation* which is used to evaluate the house with HOT2000.

(3) *Evaluation summary* which is used to summarize the evaluation results from MOIST3.0 and HOT2000.

(4) *Exit evaluation* which is used to terminate the evaluation and return to the general data input.

Once the user has clicked the button MOIST3.0 or HOT2000 Evaluation, the system invokes these two simulation programs automatically. Moreover, the input data imported from IFC or input manually can also be converted into the input files of these two programs. After the evaluation with MOIST3.0 and HOT2000 is finished, the user can click the *evaluation summary* and view the evaluation results. If the evaluation results do not meet the requirements in BEPAT, the user can go back to the general data input and revise the input data.

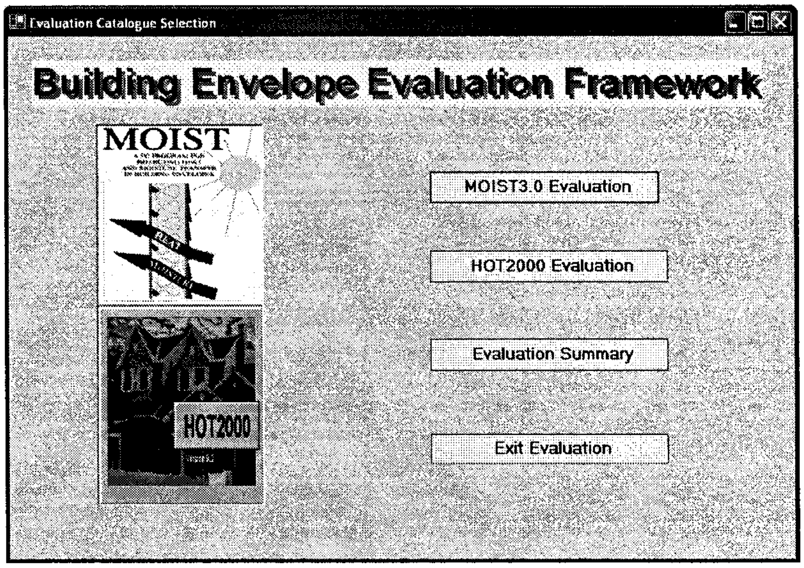


Figure 5-17. Interface for Selecting Simulation Program

5.5.2. MOIST 3.0 Linkage

The process of linking the MOIST3.0 with the framework is one of the main tasks in developing the evaluation framework. It involves mapping input data from the memory, overwriting the input files, invoking the program, terminating the application, and extracting the data from the output files. The background information about MOIST3.0 can be found in Chapter 2. This chapter focuses on the development of the MOIST3.0 linkage in the framework.

Sub-programs in MOIST3.0

The MOIST3.0 analyzes a building assembly by invoking three sub-programs, MOIST.EXE, ANALYZE.EXE, and MSTRST.EXE.

(1) MOIST.EXE is a Visual Basic program module that performs the input or output processing in Window environment for the MOIST program. It is a window interface which has functionalities such as saving user's input data to a set of input files, running the program, and displaying the results from a set output files.

(2) ANALYZE.EXE is a FORTRAN program module that performs the heat and moisture transfer analysis. It loads the input files, performs heat-moisture calculations, and saves results to the output files.

(3) MSTRST.EXE is a Visual Basic program module that resets the input files to the original state as installed. It overwrites the input files with the original back up files (*.bck) which are the original files when the MOIST program are installed.

Normally, the users use the ordinary MOIST3.0 in Window environment by invoking the MOIST.EXE. Then, they invoke the MSTRST.EXE and ANALYZE.EXE manually by clicking the buttons in the graphical interface. In other word, the process of analyzing the assembly involves the human interference. Therefore, the MOIST3.0 in Window environment cannot be used as the linking program in the system. ANALYZE.EXE, on the other hand, can carry out the whole process of analyzing the assembly without the interaction with the users. Therefore, it can be used as the linking program in the system. As for the MSTRST.EXE, the system has a mechanism to invoke it first before the ANALYZE.EXE runs. This will guarantee the ANALYZE.EXE runs with the correct initial parameters.

Input and Output Files in MOIST3.0

The data input and output in MOIST3.0 are implemented in a file system, which consists of nine input files, as shown in Table 5-2, and ten output files, as shown in Table 5-3. Since these files are in ASCII format, it is very convenient for the system to transfer input data by overwriting the input files and extract output data by reading the output files. As mentioned in Chapter 4, the data flow in MOIST3.0 is the following:

(1) Inputting data from the *General Data Input* or importing data from an IFC file.

(2) Mapping input data into the memory by the *Data Transfer Class*.

(3) Transferring data from *Data Transfer Class* to MOIST3.0 input files by overwriting the input files.

(4) Invoking ANALYZE.EXE.

(5) Transferring the calculation results to MOIST3.0 output files by overwriting the output files.

(6) Extracting the data in the output files to the postprocessor.

Table 5-2. Input Files in MOIST3.0

File Name	File Description
PDATA.DAT	Property Data of the Materials
DEFSET.DAT	General Input Parameters
CONST.DAT	Building Construction Data
PEROD.DAT	Analysis Intervals
INDOOR.DAT	Indoor Parameters for Calculating Variable Indoor RH
OPTS.DAT	Analysis Options
UNITS.DAT	Selected Systems of Units
CHECK.DAT	Diagnostic Parameters
OUT.DAT	Output Selection

Table 5-3. Output Files in MOIST3.0

File Name	File Description
RESULTS.BND	Boundary Conditions for a Simulation
RESULTS.MC	Instantaneous layer moisture contents
RESULTS.SMC	Instantaneous surface moisture contents of construction layers
RESULTS.SRH	Instantaneous surface relative humidities of construction layers
RESULTS.ARH	Weekly-average surface relative humidities of construction layers
RESULTS.OUT	Summary of all input data for a MOIST project
RESULTS.AQ	Boundary heat fluxes
RESULTS.MF	Boundary moisture fluxes
RESULTS.AR	Time-average thermal resistance of construction
RESULTS.AST	Time-average surface temperature of construction layers

Tables 5-2 and 5-3 show the file name and description of the input and output files in MOIST3.0. Appendix C lists some sample input and output files.

Interface Design for MOIST3.0 Evaluation

The interface of MOIST3.0 evaluation is used to control the process of the evaluation with MOIST3.0. It has two tab pages: *default calculation*, as shown in Figure 5-18, and *customized calculation*, as shown in Figure 5-19.

(1) *Default calculation*. The MOIST3.0 analyzes the assembly in default values. It is recommended for common users who have no professional knowledge of MOIST3.0. As shown in Figure 5-18, it has the following functions:

- By clicking the button *extract general input data*, the data input from the user manually or imported from the IFC file can be written into the input files of MOIST3.0.
- By clicking the button *calculation with MOIST3.0*, the MOIST3.0 can be invoked and terminated automatically.

- By clicking the button *display default values*, the default parameters used in the analysis can be shown in the display window on the right hand side of the interface.
- By clicking the button *display evaluation summary*, the evaluation results of MOIST3.0 can be shown in the display window on the right hand side of the interface.
- By clicking the button *exit MOIST3.0*, the evaluation with MOIST3.0 is terminated.

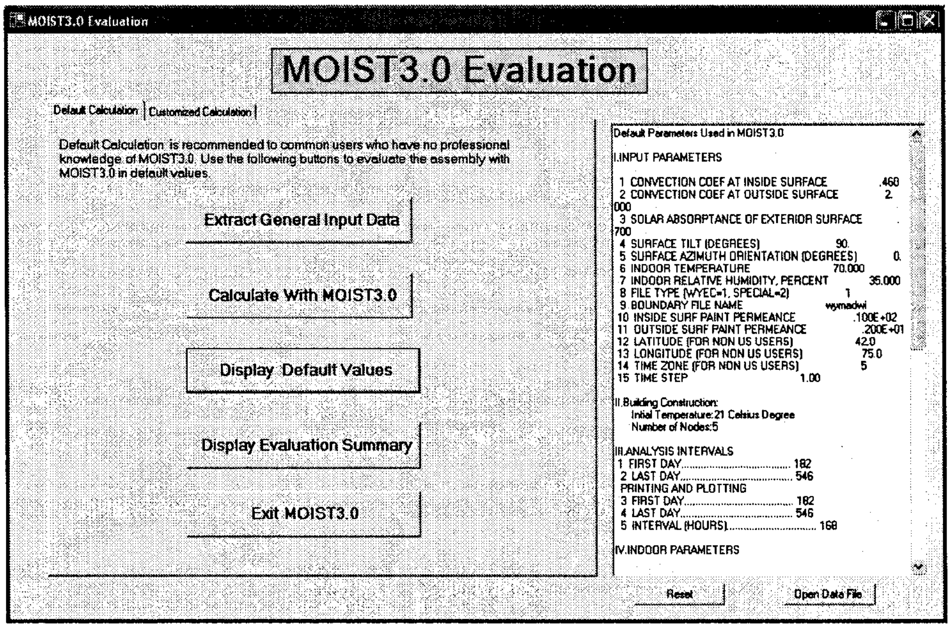


Figure 5-18. MOIST3.0 Evaluation with Default Values

(2) *Customized calculation.* The MOIST3.0 analyzes the assembly with customized values. It is recommended for expert users who have professional knowledge of MOIST3.0. As shown in Figure 5-19, it has the following functions:

- Customizing the parameters. There are five options for the user to modify the parameters of MOIST3.0. So far, only Option 1 and Option 2 are implemented.

- Option 1. By clicking the button *customize material database*, in Figure 5-19, a new interface will appear in the screen, as shown in Figure 5-20. The user can use this interface to display all materials and their properties, modify the property of a material, and add the property of a new material which is inserted in the General Data Input.
- Option 2. By clicking the button *customize input parameters*, in Figure 5-19, a new Window interface will appear in the screen, as shown in Figure 5-21. The user can use this Window interface to modify the parameters such as the convection coefficient on inside and outside surface, solar absorptance of exterior surface, time zone, latitude, longitude, and time step.

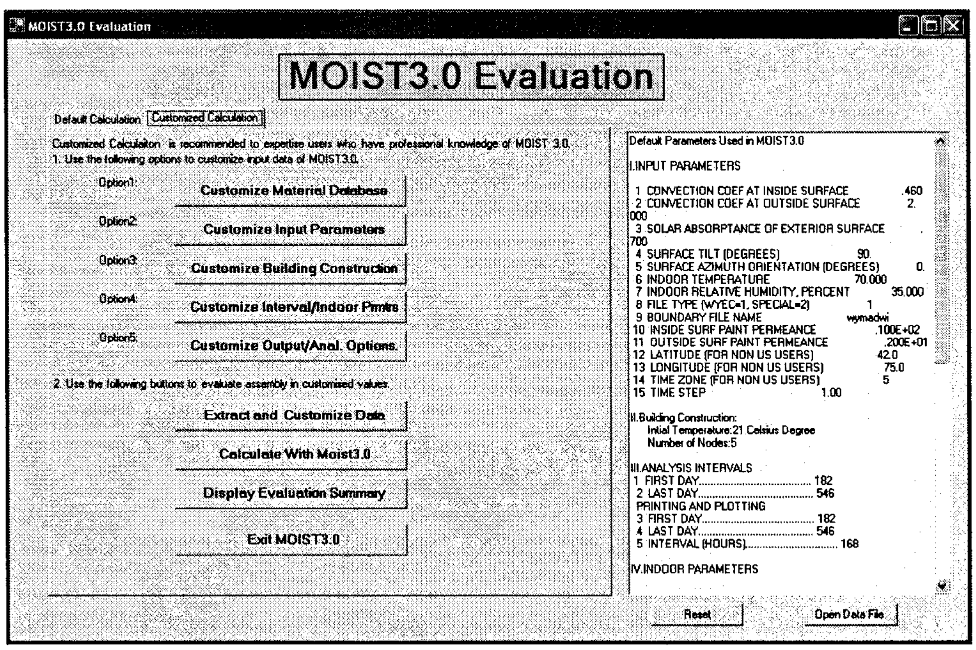


Figure 5-19. MOIST3.0 Evaluation with Customized Values

- By clicking the button *extract and customize data*, in Figure 5-19, the data input from the *General Data Input* or imported from the IFC file can be written into the input files of MOIST3.0. Moreover, the customized parameters in Figures 5-20, 5-21 are also saved into the system.
- Other functions, which are initiated with buttons: *display the default values*, *analysis summary*, and *exit MOIST3.0*, are exactly the same as that in default calculation in Figure 5-18.

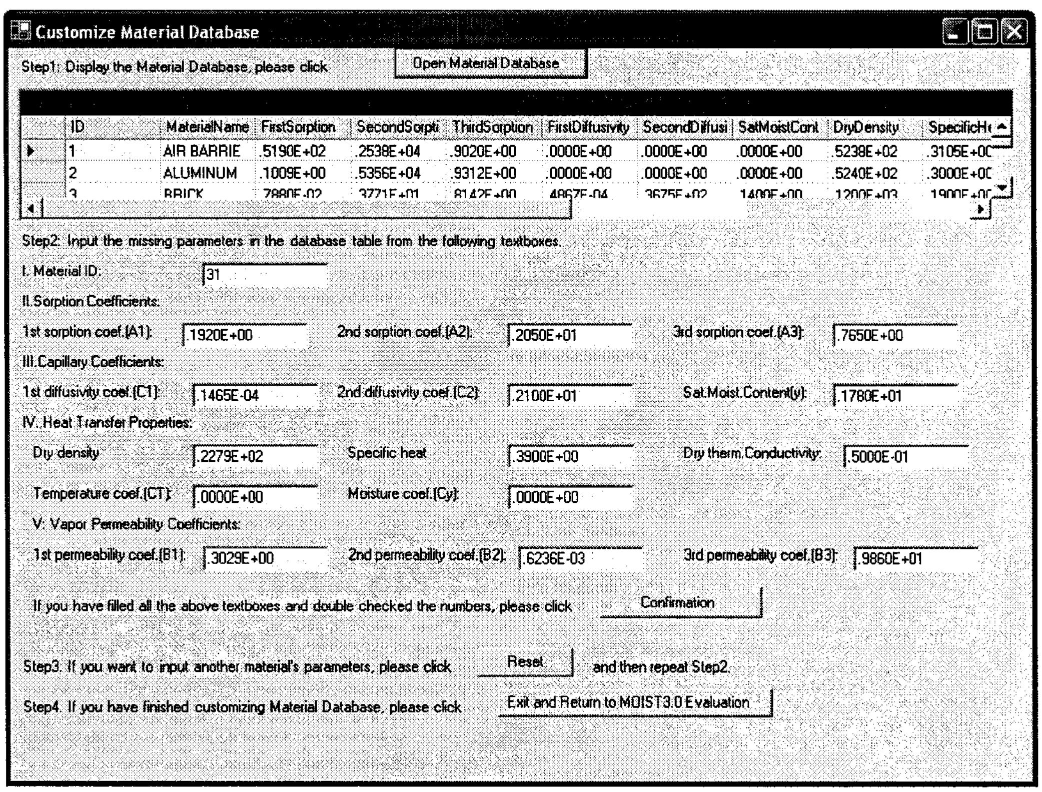


Figure 5-20. Interface of Customizing the Material Database

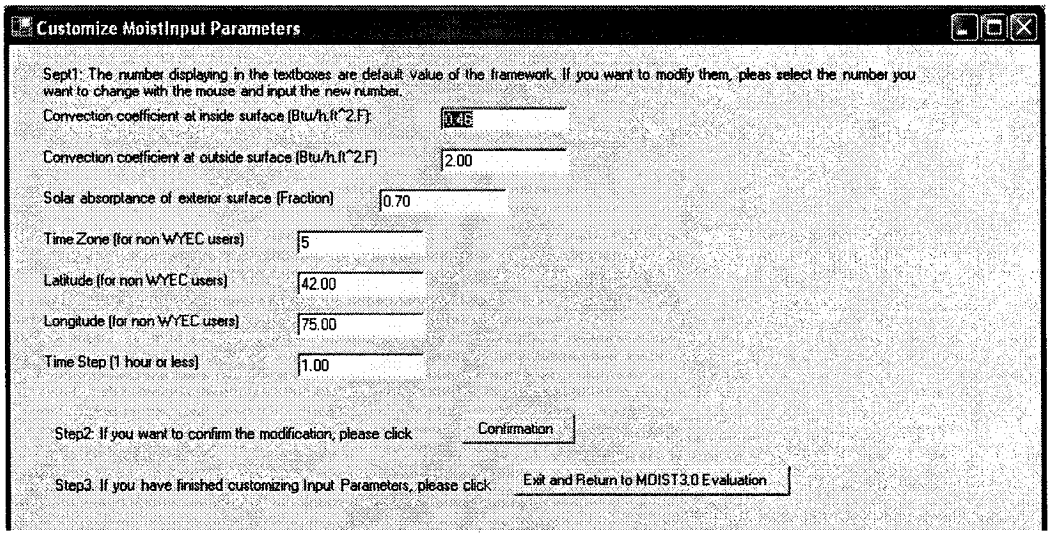


Figure 5-21. Interface of Customizing the Input Parameters

5.5.3. HOT2000 Linkage

Besides the MOIST3.0, the process of linking the HOT2000 with the framework is another main task in developing the evaluation framework. Like the MOIST3.0, this process also involves mapping input data from the memory, extracting R-value from the output file of MOIST3.0, overwriting the input files, invoking the .exe program, terminating the application, and extracting the data from the output files. The background information about HOT2000 can be found in Chapter 2. This chapter focuses on the development of the HOT2000 linkage in the framework.

In addition, HOT2000's Windows version can be downloaded freely in the public domain, but the system could not link to it directly because the Windows version has no API. However, NRCan has provided two methods to link to HOT2000. One is HOT2000

API with DLL (Dynamic Link Library) (Haltrecht et al., 1999); the other is BATCH HOT2000 (Bradley, 2003). Both of them can be easily linked into to the system. The following is the brief introduction about these two linking methods.

<u>Link to HOT2000 with DLL</u>

DLL is a special type of Windows program containing functions that can be called by other programs or resources. It is a file of functions compiled, linked, and saved separately from the processes that use them. The application links to the functionality in the library file when the application is run. The library file remains a separate file referenced and called by the application. DLL can be used in any Windows programming language environments.

The HOT2000 API with DLL is one way to use the core HOT2000 engine. Developers could use it to open and modify HOT2000 house files, perform calculations, and produce reports. There are a lot of public functions and variables for users to manage the HOT2000 data in this API. Furthermore, it is possible to use it with Visual Basic, FORTRAN or Pascal language and encapsulate the functionality of the core as a class in C++ language (Haltrecht et al., 1999).

Link to HOT2000 with Batch Version

The BATCH HOT2000 version is written with standard FORTRAN 77. Its original purpose is to estimate the space heating requirements for a large number of houses. Both its input and output data are written in ASCII files which may be edited in any compatible text-editing programs. The output report could be saved as a file in the comma-separated-values (CSV) format and may contain user selected summary input and calculated results fields. The CSV file is compatible with spreadsheet formats such as Excel, Lotus, Quatro, and so on. There are three important files in running the Batch Version: Job Control File (*.JCF), Job Control Record (*.JCC), and Job Input File (Bradley, 2003). The input data for a house are stored in a pair of files, *.V71 and *.V80. The data input in the preprocessor of the system are transferred to the HOT2000 by overwriting the *.V71 and *.V80 input files. The system invokes the BATCH HOT2000 program, and then calculates and outputs the result. In the project, the system links to HOT2000 by BATCH HOT2000 since it is simpler than by the DLL in terms of programming.

<u>Input and Output Files in HOT2000Batch Version</u>

Like MOIST3.0, the data input and output in HOT2000 are also implemented in a file system, which consists of four input files, *.JCF, *.JCC, *.V71, and *.V80 and three output files, Export.CSV, RPT, Scree. Like MOIST3.0, these files are in ASCII format. Therefore, it is very convenient for the system to transfer input data by overwriting the input files and extract output data by reading the output files. Some input and output files are listed in Appendix D. The followings are just their definitions:

(1) Job Control File (*.JCF). It is a list, one name per line, of the names of files to be processed by the program. The Job Control Record is the first file named in the Job Control File.

(2) Job Control Record (*.JCC). It defines the names of the program data files, alternate search paths which the program will use to locate data files, and a "command" line to enter an optional command. This record will also include a list of the fields selected for spreadsheet output.

(3) Job Input Files (A pair of *.V71 and *.V80). They consist of any number of data sets, i.e. house description files. Each house description file is an ASCII file.

(4) Export. CSV. It is an output file in spreadsheet compatible format, CSV (Comma Separated Value)

(5) RPT. It is an output file in text format.

(6) Scree. It is a log file which records the running process of the HOT2000 Batch.

As mentioned in Chapter 4, the process of data flow in HOT2000 is the following:

(1) Inputting data from the general data input or importing data from an IFC file.

(2) Mapping input data into the memory with the *Data Transfer Class*.

(3) Transferring data from *Data Transfer Class* to HOT2000 input files by overwriting the input files (*.V71 and *.V80).

(4) Invoking the HOT2000 Batch (*.BAT).

(5) Invoking the control and input files (*.JCF, *.JCC, *.V71, and *.V80).

(6) Transferring the calculation results to HOT2000 output files by overwriting the output files (Export.CSV, RPT, and Scree).

(7) Extracting the data in the output files to the postprocessor.

Interface Design for HOT2000 Evaluation

The interface of HOT2000 Evaluation is used to control the process of the evaluation with HOT2000. It has two tab pages: default calculation and customized calculation. The customized calculation has not been implemented at this moment. The following is the introduction of the default calculation, as shown in Figure 5-22.

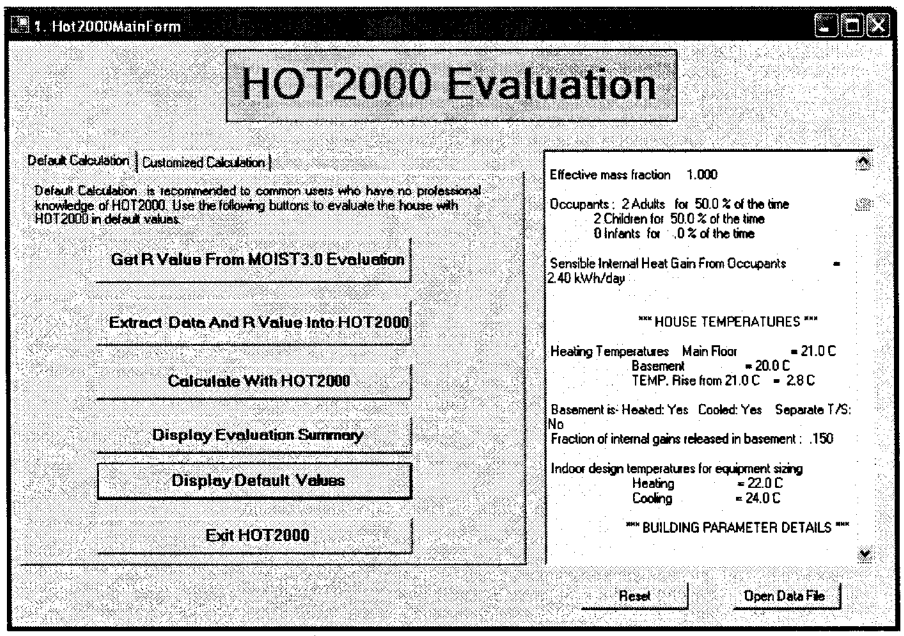


Figure 5-22. HOT2000 Evaluation with Default Calculation

(1) By clicking the button *get R-value from MOIST3.0 evaluation*, the R-value of the assembly calculated by MOIST3.0 is extracted from the output file of the MOIST3.0.

(2) By clicking the button *extract data and R-value into HOT2000*, the data input from the users manually or imported from the IFC file can be written into the input files of HOT2000.

(3) By clicking the button *calculate with HOT2000*, the HOT2000 can be invoked and terminated automatically.

(4) By clicking the button *display evaluation summary*, the evaluation results of the HOT2000 can be shown in the display window on the right hand side of the interface.

(5) By clicking the button *display the default values*, the default parameters used in the analysis can be shown in the display window on the right hand side of the interface.

(6) By clicking the button *exit HOT2000*, the evaluation with HOT2000 is terminated.

5.5.4. The Data Transfer between HOT2000 and MOIST3.0

Data transfer between simulation programs is one important feature to be developed in the framework. If this process can be achieved, the simulation programs linked in the system can work together and provide the assembly and house with a more comprehensively evaluation. As mentioned in the above section, during evaluation with HOT2000, the framework gets the R-value from MOIST3.0 as the input data in HOT2000. This is one sample to transfer data between linked simulation programs.

In HOT2000 Batch, all non-window structural components such as wall and roof are described by one of three methods. These three methods require the user to specify a few values such as height and perimeter for walls. The structural codes for components are expressed in 10 digit numbers. These three methods are as the following:

(1) Method 1. The user enters ten 0 and an R-Value (R or RSI) for the component. Example: code: 0000000000, RSI: 7.04. This method is simple, but the R-value is not accurate because the actual R-value depends on all the various layers in the

component and on thermal bridging, unless the user has calculated these factors by some other method.

(2) Method 2. The user uses the HOT2000's menus to define a structural code for a component. A structural code defines each layer of the component, including stud or joist type, dimensions and spacing, insulation, interior, exterior, and corners. Example: code: 1211321140, RSI: calculate Decode: 1---wall, 2---wood frame, 1---size 38x140mm, 1---spacing 400mm, 3---insulation layer1 RSI3.4 R20Batt, 2---insulation layer2 38mm EPSII, 1---waferboard/OSB 9.5mm, 4---brick, 0---2 studs. In this method, the HOT2000 calculates the actual (effective) R-value. However, not all possible construction types are covered by the menus.

(3) Method 3. The user defines components. This method uses digital numbers like that in structural code in Method 2 to describe the structure of a component, but the format is different from that in Method 2. In this method, the HOT2000 calculates the actual (effective) R-value. Moreover, it is very easy to coordinate with the material input of other simulation programs. However, the format in Method 3 is very complicated and difficult to implement.

In this prototype system, for the sake of simplification, Method 1 is used to express the wall or roof in HOT2000. This is why the R-value is transferred from the output file in MOIST3.0 to the input file in HOT2000.

5.6. Implementation of the Postprocessor

As presented in Chapter 4, the *postprocessor* is a functional unit in the system that processes the performance values of the building envelope. So far, only the performance values which are generated automatically by simulation programs in the *application integrator* are evaluated by the system. The *postprocessor* is used to retrieve performance evaluation criteria from the rule database, compare the performance value with the retrieved criteria, and create a final report to the user. The rule database contains rules extracted from the BEPAT.

5.6.1. Evaluation Summary

In the *application integrator*, each application evaluates the house from its respective aspects. For instance, HOT2000 is focusing on energy simulation in a house, and MOIST3.0 is focusing on thermal and moisture analysis in a wall or roof. However, the output of these applications is very complicated and large in size. Therefore, it is very difficult for a user who has no expertise in these applications to interpret the results. The evaluation summary is to simplify the results by selecting the contents that the user may be most interested in. These selected results will be used to compare with a set of criteria which will be discussed in the next section.

5.6.2. Comparison between the Evaluation Summary and the Criteria from BEPAT

The BEPAT, as described in Chapters 2 and 4, is used to evaluate the results in this research. With BEPAT, the system can search for the rules corresponding to the results in the evaluation summary; then the results are checked against the retrieved rules. If the results meet the requirement of the rules, they are accepted and the system will output the final report. Otherwise, the user can go back to the very beginning of the system and revise the input data. During this process, knowledge-based code checking with IFC technology should be applied. So far, the comparison is limited to one part of BEPAT by extracting the rules (clauses) within the system, and the knowledge-based code checking is not included in the system, so the above process is only at the conceptual stage and the implementation of the full concept will be integrated into the framework in the future. Figures 6-6, 6-7, 6-8, and 6-9 are the sample interfaces of the evaluation summary and the compared results with part of BEPAT (as will be explained in the case study in next chapter). There are three options for the user to view the evaluation summary:

(1) MOIST3.0 summary. The user can view information including the R-value of the wall and comparison of this result with the respective criterion. For wood-based material, the summary shows the range of moisture content in the layer. If moisture contents are greater that 19%, the summary indicates the duration in which the moisture contents are greater than this value.

(2) HOT2000 summary. The user can view information including: name of the weather city, building surface area, annual heating degree days, annual heating energy consumption, annual cooling energy consumption, annual DHW (Domestic Hot

Water) energy consumption, annual appliance energy consumption, and annual energy total consumption.

(3) Full evaluation summary. The user can view information including that in 1 and 2.

Chapter 6

Case Study

6.1. Introduction

The previous chapters have discussed the literature about the building envelope performance evaluation, the concept and application of the IFC, and the design and the implementation of the evaluation framework. Based on these discussions, this chapter will demonstrate the application of IFC and the integrated evaluation framework with a case study. This case study includes:

(1) Importing geometry and material layer data from an IFC file using Eurostep Active Toolbox,

(2) Evaluating the hygrothermal and energy performances with the framework.

The objective of this case study is to prove that the framework can import data from the IFC model and evaluate the energy, thermal and moisture performance concurrently by the proposed integrated approach.

6.2.Case Study Description

The selected house is one of ten houses designed for the Advanced House Program in Canada, which showcases the Canadian innovations and affirms the commitment that fosters a greener environment. It is called the NOVTEC Advanced House and was built in the 1990's.

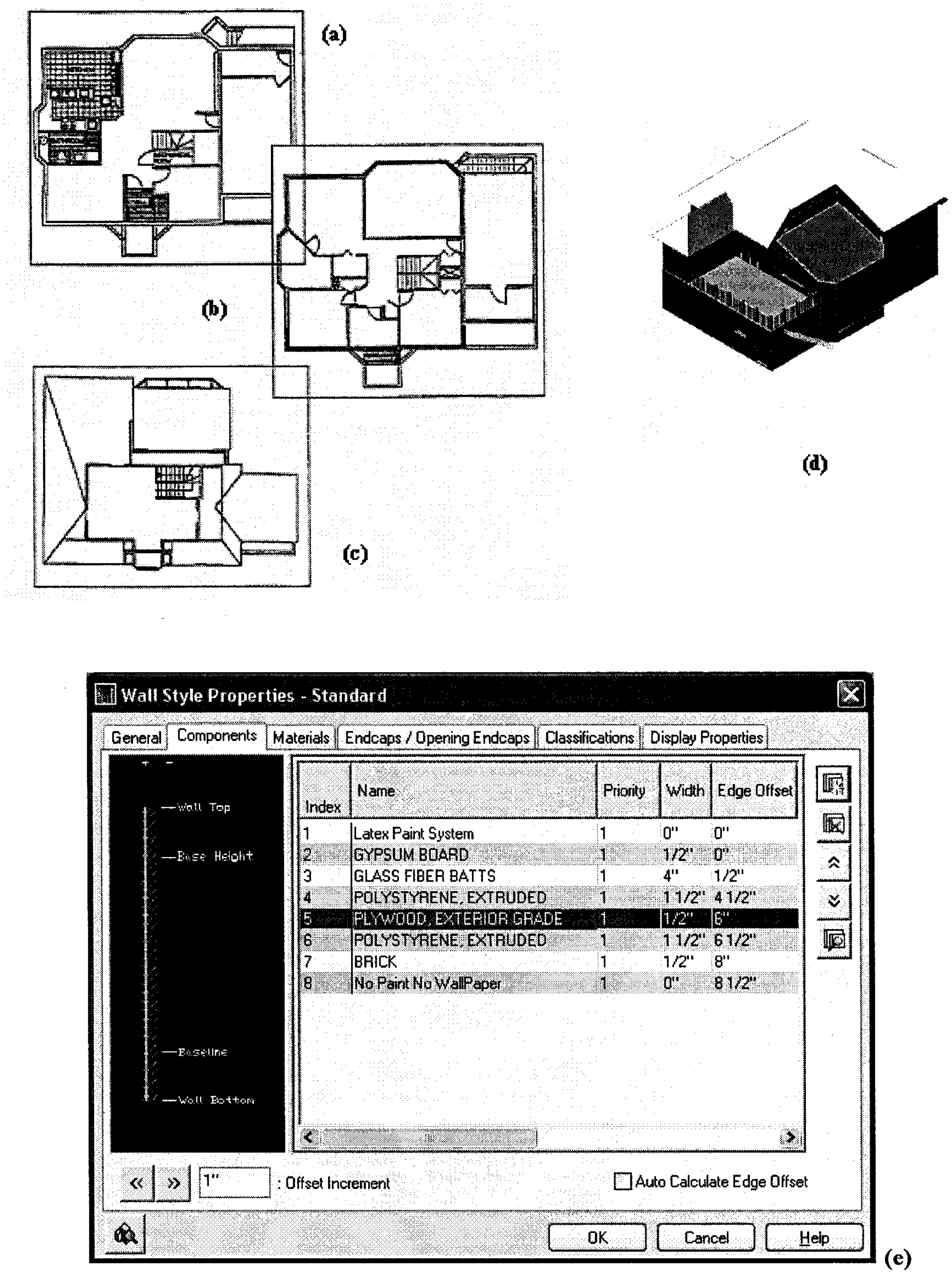


Figure 6-1. Plans of the (a) ground floor, (b) first floor, and (c) mezzanine floor of the NOVTEC Advanced House (Gerbasi, 2000), (d) 3D model of NOVTEC house created with Architectural Desktop 2005, (e) Cross section of the exterior wall in CAD model created with Architectural Desktop 2005

The house is a detached single-family house with 2 and half stories and 3 bedrooms. It is located in Laval, Quebec, near Montreal. Except for the garage, the house has 222m^2 of heated floor area and a total volume of 602 m^3. The plan drawings of each floor are as shown in Figure 6-1 (a-c), and the exterior wall with a cross section of EIFS (Exterior Insulation and Finish System) are illustrated in Figure 6-2. Before the occupants moved in the house, the energy performance of the house had been monitored. For privacy reason, the energy consumption for Domestic Hot Water (DHW) was not monitored in the field test system. Therefore, there are no DHW data available in this case study. The data from the field tests were collected over a 21-month period from November 1993 to July 1995 (Gerbasi, 2000).

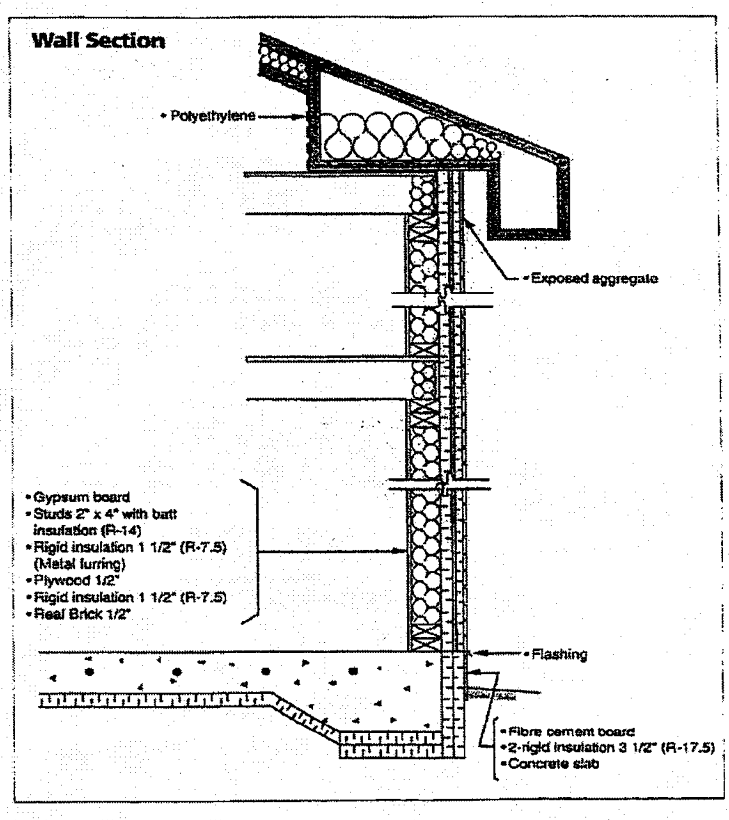


Figure 6-2. Cross-section of the Exterior Insulation and Finish System (Gerbasi, 2000)

6.3. Importing Geometry and Material Layer Data from an IFC File

In Chapter 5, the interface and the algorithm of importing the geometry and material layer data from an IFC file have been introduced. With this interface and algorithm, the geometry and material layer data of a building in an IFC file can be imported into the system by the Eurostep Active Toolbox. This section shows how to carry out this process in the case study.

6.3.1. Generating IFC File with Architectural Desktop 2005

As mentioned in Chapter 5, data transfer from CAD drawings to IFC files could be achieved automatically by IFC-compatible CAD applications. The AutoDesk's Architectural Desktop with its IFC2x Utility (Inopso GmbH, 2005) and Graphisoft's ArchiCAD with its add-on interface (Graphisoft, 2005) are two typical applications that can generate an IFC file from the CAD drawings. In this project, Architectural Desktop 2005 with its IFC2x Utility is used as the tool to automatically generate an IFC file from the CAD drawings. A 3D model of the house described in section 6.1 was created with this application as shown in Figure 6-1 (d), and the cross section of the exterior wall in CAD model was also created with this application as shown in Figure 6-1 (e). In addition, a part of the IFC text file generated by the IFC2x Utility is shown in Appendix E.

6.3.2. Imported Geometry and Material Layer Data

After the user applies the procedures described in Chapter 5, the geometry and material data are imported. Figures 6-3 (a) and (b) are the screen shots of importing the IFC data of the NOVTEC house. As shown in Figure 6-3 (a), geometry data, such as the number of stories, and the total length and average height of the exterior walls in each floor, are imported in the system and are transferred into HOT2000. Meanwhile, as shown in Figure 6-3 (b), material layer data, such as interior and exterior finishes, material name and thickness of each material layer of the exterior walls, are imported in the system and are transferred into MOIST3.0. The geometry data displayed in Figure 6-3 (a) are listed in Table 6-1, and material layer data displayed in Figure 6-3 (b) are listed in Table 6-3.

Table 6-1. Geometry Data Imported from the IFC File

Story NO.	Total Length of Ex. Wall (in.)	Average Height of Ex. Wall (in.)
1	1547	121
2	1541	98
3	1191	50

6.4. Evaluating the Hygrothermal and Energy Performances with the Framework

6.4.1. Evaluating Approach

In this case study, the data about the geometry and the mechanical system are exactly the same as that in the NOVTEC house, but the wall cross-section has two variants: the *Non-efficient Wall* and the *Efficient Wall*. The purpose of selecting these variants is to demonstrate the impact on the performance values by the different wall layer compositions. The *Non-efficient Wall* is derived from the sample wall in MOIST3.0 program. The *Efficient Wall* is the same as the original Exterior Insulation Finish System (EIFS) wall in the NOVTEC house. The layer information of these walls is listed in Tables 6-2 and 6-3 and shown in Figures 6-4 and 6-5.

Table 6-2. Layer Information of the *Non-efficient Wall*

Layer Position	Material	Thickness (in.)
Interior Finish	Latex Paint	N/A
Layer 1	Gypsum Board	0.5
Layer 2	Glass Fiber Batts* (R11)	3.5
Layer 3	Fibre Board Sheathing	0.5
Layer 4	Sugar Pine	0.5
Exterior Finish	No paint no wall paper	N/A

*Non-storage Material: is the insulation material with low density (e.g. glass fiber blanket insulation and cellulose insulation) and cannot store heat and moisture.

Table 6-3. Layer Information of the *Efficient Wall*

Layer Position	Material	Thickness (in.)
Interior Finish	Latex Paint	N/A
Layer 1	Gypsum Board	0.5
Layer 2	Glass Fiber Batts* (R14)	4
Layer 3	Extruded Polystyrene* (R7.5)	1.5
Layer 4	Plywood	0.5
Layer 5	Extruded Polystyrene* (R7.5)	1.5
Layer 6	Brick	0.5
Exterior Finish	No paint no wall paper	N/A

*Non-storage Material: the same as Table 6-2

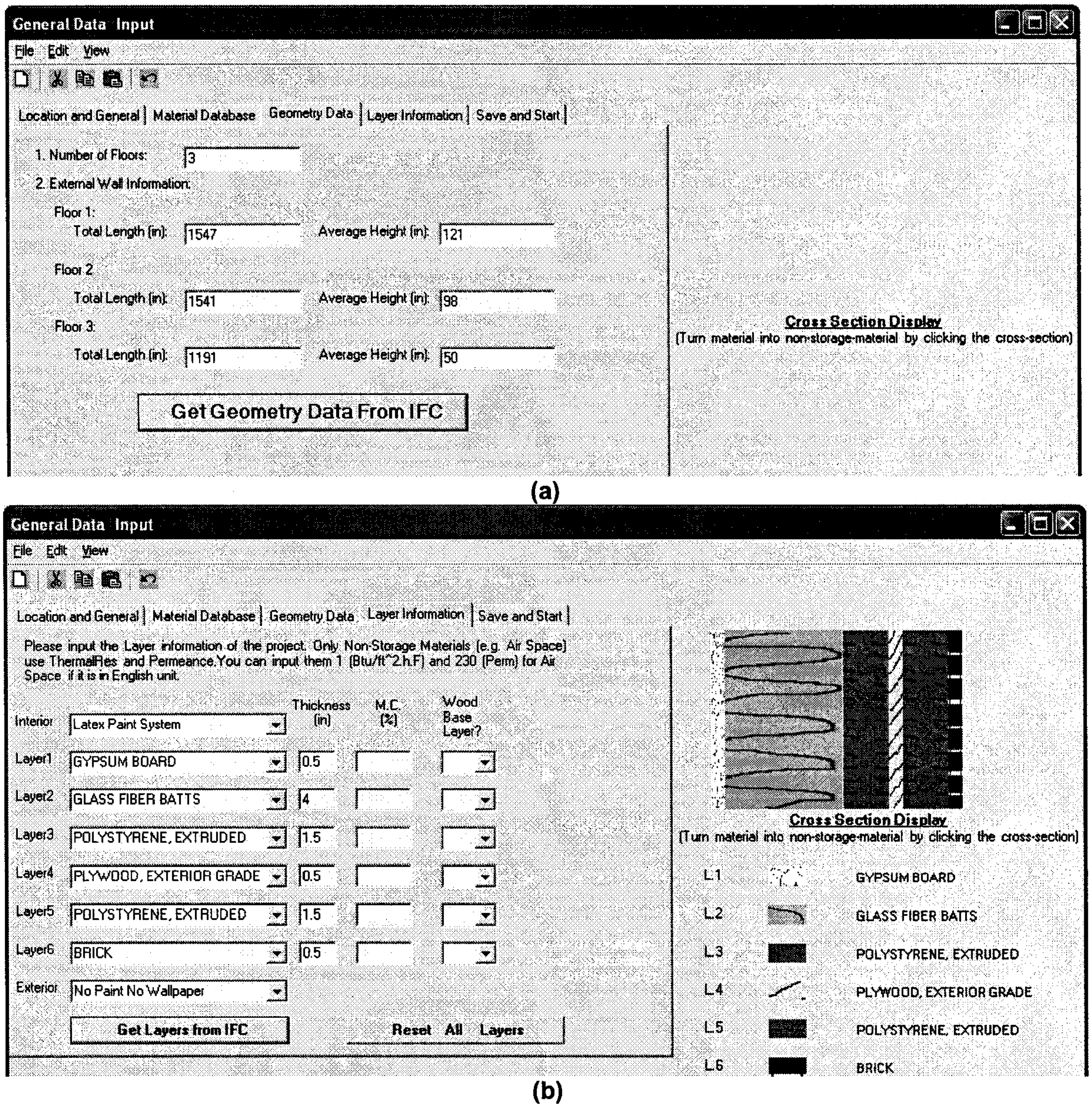


Figure 6-3. (a) Geometry data of the exterior walls imported into the system

(b) Material layer data imported into the system

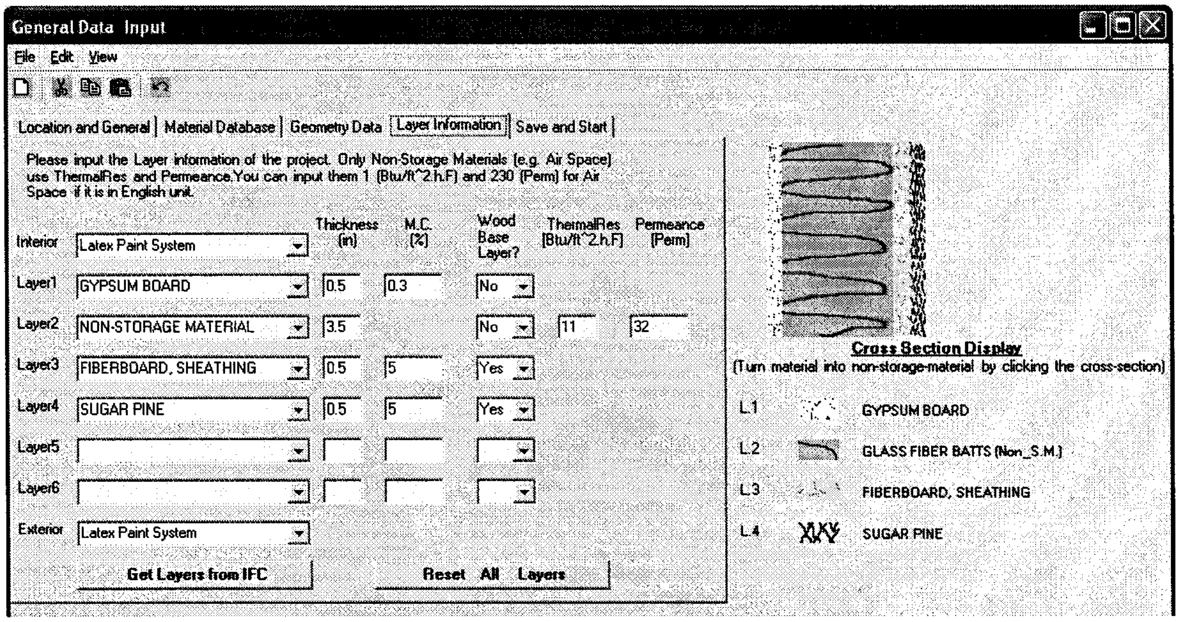


Figure 6-4. Layer Information of the *Non-efficient Wall*

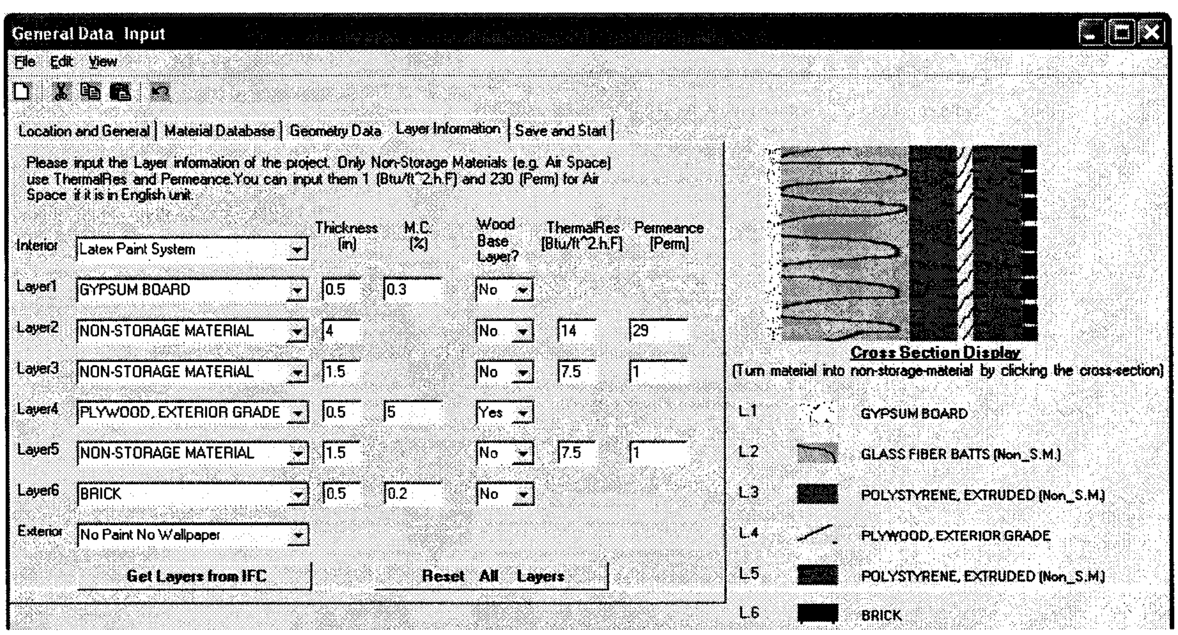


Figure 6-5. Layer Information of the *Efficient Wall*

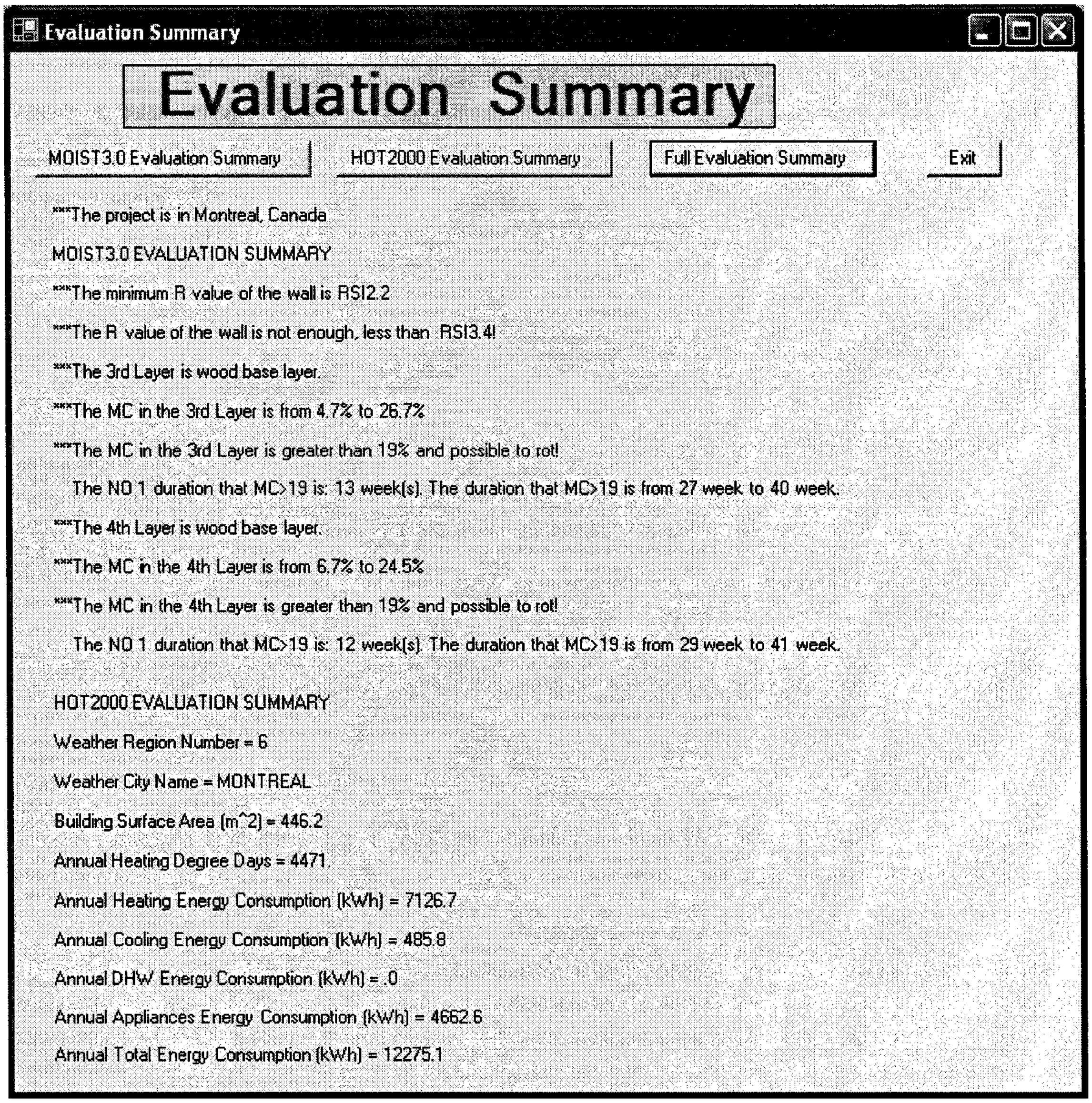


Figure 6-6. Evaluation Summary of the *Non-efficient Wall* in Montreal

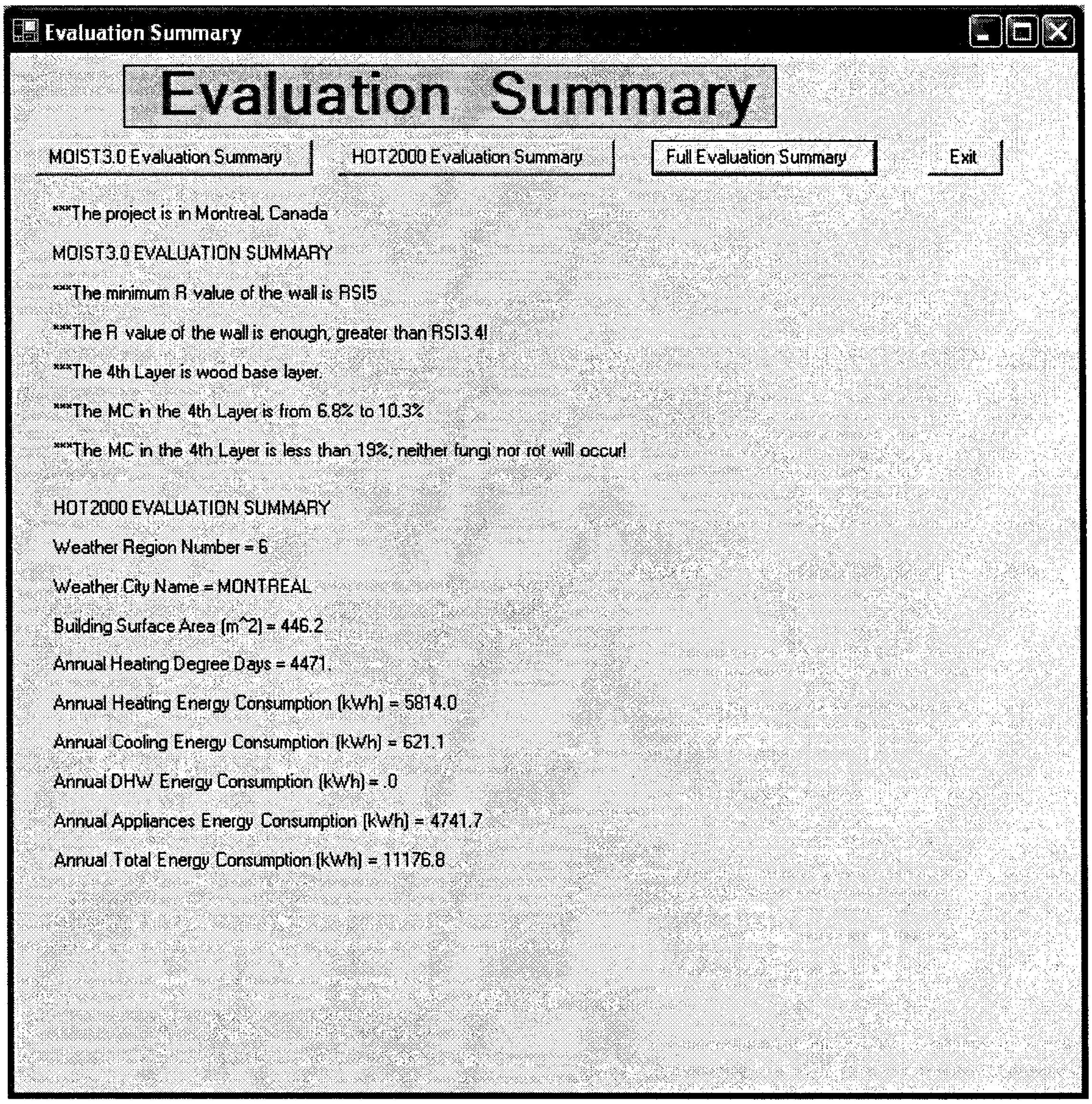


Figure 6-7. Evaluation Summary of the *Efficient Wall* in Montreal

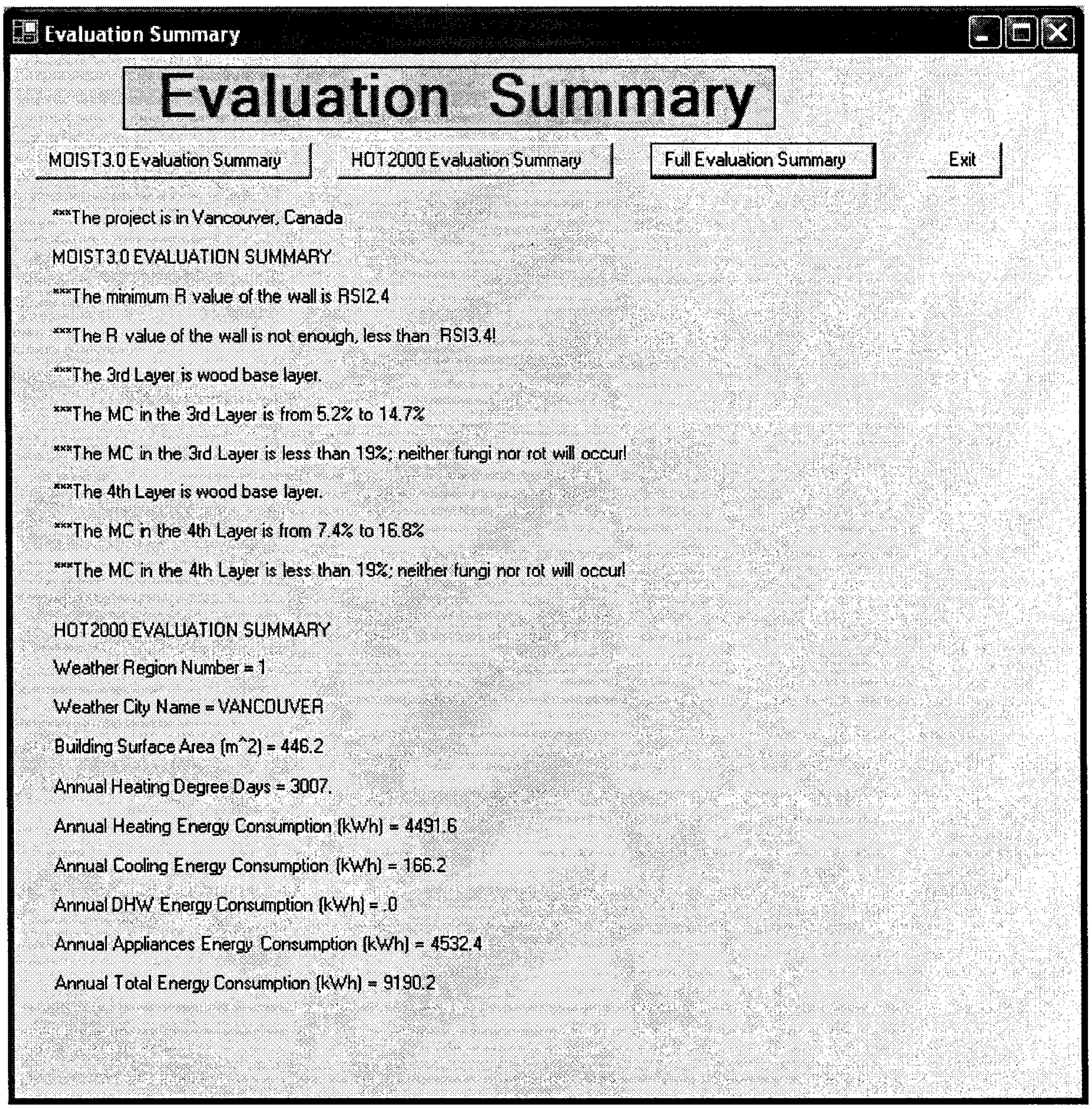


Figure 6-8. Evaluation Summary of the *Non-efficient Wall* in Vancouver

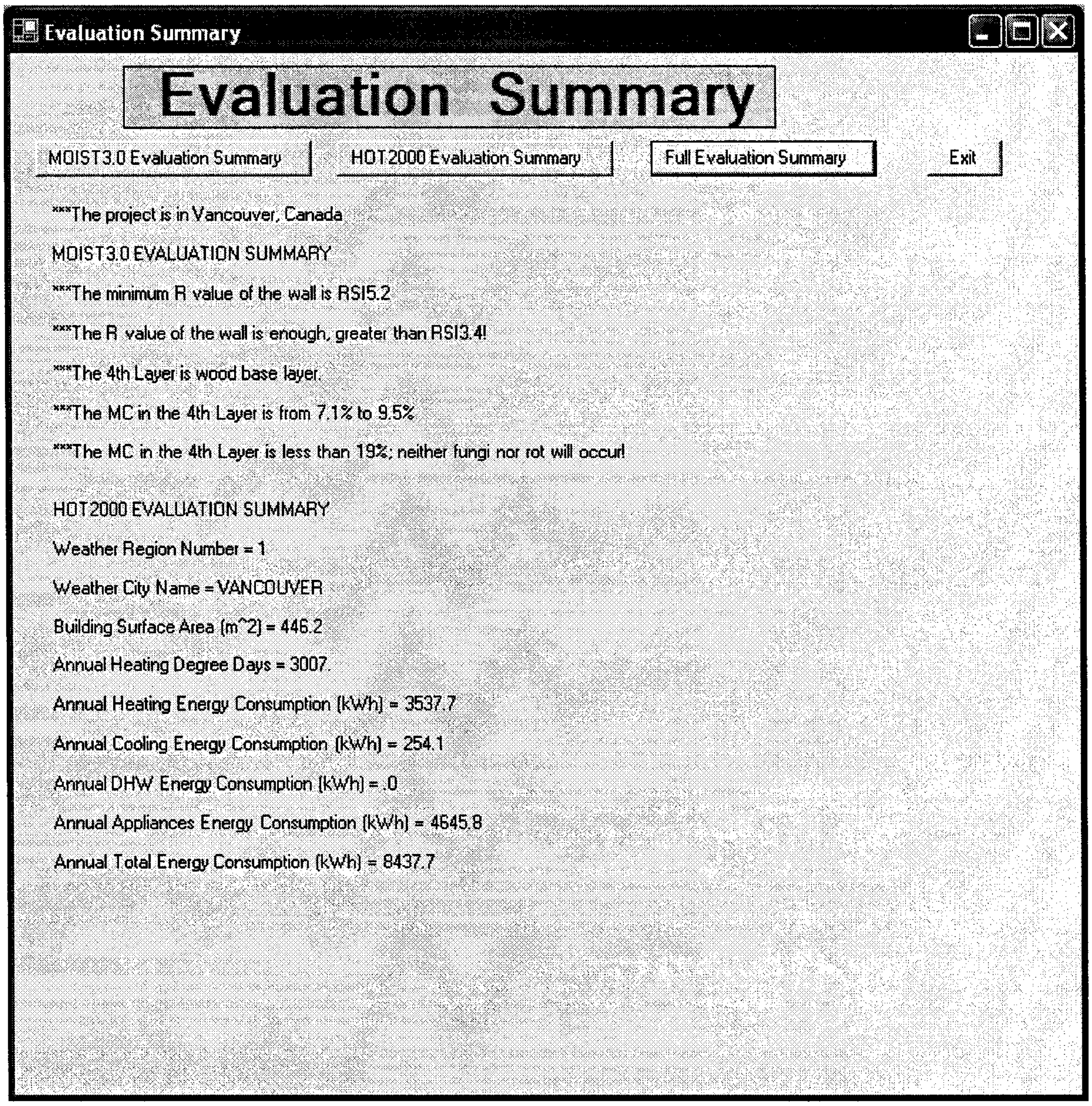


Figure 6-9. Evaluation Summary of the *Efficient Wall* in Vancouver

Table 6-4.
Evaluation Summary of the NOVTEC House with the *Non-efficient Wall* in Two Locations

Location			Montreal	Vancouver
Minimum R Value (RSI)			2.2	2.4
Wood-based layer	Fibre Board Sheathing	MC (%)	4.6 ~ 26.7	5.2 ~ 14.7
		Duration (MC>19%)	13 weeks	N/A
	Sugar Pipe	MC (%)	6.7 ~ 24.5	7.4 ~ 16.8
		Duration (MC>19%)	12 weeks	N/A
Annual Heating Degree Day (°C-days)			4471	3007
Annual Heating Energy Consumption (kWh)			7126.7	4491.6
Annual Cooling Energy Consumption (kWh)			485.5	166.2
Annual Appliance Energy Consumption (kWh)			4662.6	4532.4
Annual Total Energy Consumption (kWh)			12275.1	9190.2
Duration of Simulation			1 year	
Thickness of the Wall (in.)			5	

Table 6-5.
Evaluation Summary of the NOVTEC House with the *Efficient Wall* in Two Locations

Location			Montreal	Vancouver
Minimum R Value (RSI)			5.0	5.2
Wood-based layer	Plywood	MC (%)	6.8 ~ 10.3	7.1 ~ 9.5
		Duration (MC>19%)	N/A	N/A
Annual Heating Degree Day (°C-days)			4471	3007
Annual Heating Energy Consumption (kWh)			5814.0	3537.7
Annual Cooling Energy Consumption (kWh)			621.1	254.1
Annual Appliance Energy Consumption (kWh)			4741.7	4645.8
Annual Total Energy Consumption (kWh)			11178.6	8437.7
Duration of the Simulation			1 year	
Thickness of the Wall (in.)			8.5	

Besides evaluating the house in the original weather location of Montreal (Laval is near to Montreal), one more city, Vancouver, is also used as an evaluation location. The purpose of evaluating the same house in two locations is to demonstrate that the performance values of a wall in different locations vary significantly. They may meet the requirement in one place but not in another. Because there are two types of walls and two locations, the total evaluation summary outputs are four. Figures 6-6, 6-7, 6-8, and 6-9 show the evaluation summaries.

6.4.2. Evaluation Summary

In order to compare the evaluation summary of the NOVTEC house in different locations and with different wall cross sections, Tables 6-4 and 6-5 list part of the performance values for the NOVTEC house with two wall cross sections in two locations. These performance values are calculated by the simulation applications, HOT2000 and MOIST3.0 (The R-values and MC are calculated by MOIST3.0, and the annual energy consumption is calculated by HOT2000). These calculations are not carried out directly in HOT2000 and MOIST3.0. Instead, they are carried out in the system by invoking these two applications. The following is the analysis of the data in these two tables.

Table 6-4 shows the evaluation summary of the NOVTEC house with the *Non-efficient Wall* in two locations. The following is the analysis to the data in this table:

(1) Due to the difference of the annual heating degree-day in the two locations (ranging from 4471 to 3007 °C-days), the values of the annual heating energy consumption of

the house are significantly different in the two locations. The value in Montreal is higher (7126.7 kWh), while the value in Vancouver is lower (4491.6 kWh). Due to the same reason, the values of the annual cooling energy consumption of the house are also significantly different in two locations. The value in Montreal is higher (485.5 kWh), while the value in Vancouver is lower (166.2 kWh). Since the regional climate only has minor influence on the use of the appliance in the house, the annual appliance energy consumption of the NOVTEC house in the two locations are slightly different, that is, 4662.6 kWh in Montreal and 4532.4 kWh in Vancouver. As a result, the difference between the annual total energy consumption of the NOVTEC house in Montreal (12275.1kWh) and in Vancouver (9190.2kWh) is mainly caused by the difference between the energy consumption of heating and cooling in these two locations.

(2) The moisture contents (MC) in the two wood-based layers (fireboard sheathing and sugar pine) are also significantly different. In Vancouver, the maximum moisture contents in these layers are 14.7 % and 16.8 %, respectively, and are also less than 19%, the threshold mentioned in the BEPAT. Thus, no rot will occur in these two wood-based layers. However, in Montreal, the maximum moisture contents in these layers are 26.7% and 24.5 %, respectively, and are greater than 19%. Moreover, the durations that the MCs are greater than 19% in these layers are 13 weeks and 12 weeks, respectively. Thus, moisture damage due to condensation may occur in these two wood-based layers in Montreal.

(3) Although the moisture content could impact the thermal resistance of the wall, the R-values of the wall in the two cities vary slightly. They are 2.2, and 2.4 RSI,

respectively, in Montreal and Vancouver. Moreover, the R-value of the wall in Montreal is 2.2 RSI, which is less than the 3.4 RSI, the threshold in the BEPAT for Montreal. Thus, a new insulation layer with a 3.4 RSI is recommended.

Table 6-5 shows the evaluation summary of the NOVTEC house with the *Efficient Wall* in two locations. The following is the analysis to the data in this table:

(1) The same reason as in Table 6-4, the values of the annual heating energy consumption of the house are significantly different in the two locations. The value in Montreal is higher (5814.0 kWh), while the value in Vancouver is the lower (3537.7 kWh). Due to the same reason, the values of the annual cooling energy consumption of the house are also significantly different in two locations. The value in Montreal is higher (621.1kWh), while the value in Vancouver is lower (254.1 kWh). Since the regional climate has minor influence on the use of the appliance in the house, the annual appliance energy consumption of the NOVTEC house in the two locations are slightly different, that is, 4741.7 kWh in Montreal and 4645.8 kWh in Vancouver. As a result, the difference between the annual total energy consumption of the NOVTEC house in Montreal (11176.8 kWh) and in Vancouver (8437.7 kWh) is mainly caused by the difference between energy consumption of heating and cooling in these two locations.

(2) The moisture contents (MC) of the wood-based layer plywood in two locations are very similar. The maximum moisture contents in this layer are 10.3%, and 9.5 %, respectively, in Montreal and Vancouver. All of them are less than 19%, the threshold

mentioned in the BEPAT. Thus, no rot will occur in this wood-based layer in two locations.

(3) For the same reason as in Table 6-4, the R-values of the wall in two locations vary slightly. They are 5.0, and 5.2 RSI, respectively, in Montreal and Vancouver. Moreover, the R-value of the wall in Montreal is 5.0 RSI, which is greater than the 3.4 RSI, the threshold in the BEPAT for Montreal. Thus, the *Efficient Wall* meets the requirement in the BEPAT.

By comparing the value of the hygrothermal and energy performance in Tables 6-4 and 6-5, one can conclude that the *Efficient Wall* is more suitable than the *Non-efficient Wall* in the case of Montreal. For example, in Montreal, the annual total energy consumption of the house with the *Efficient Wall* is 11176.8 kWh, which is less than that with the *Non-efficient Wall* 12275.1 kWh. In addition, the maximum MC of the wood-based layer for *Efficient Wall* is 10.3%, which is less that that (26.7%) for *Non-efficient Wall.* Moreover, the *Efficient Wall* can be used both in Montreal and in Vancouver because no MCs in the wood-based layers are greater than 19%. However, the *Non-Efficient Wall* can only be used in Vancouver but not in Montreal because the MCs of the wood-based layers are greater that than 19% (26.7% in fibre board sheathing and 24.5% in sugar pipe). Finally, in terms of cost, the cost of the *Efficient Wall* is more expensive than that of the *Non-efficient Wall* since their thickness are 8.5 inches (21.59 cm) and 5 inches (12.7 cm), respectively. However, in terms of lifecycle cost and environment protection, the house with *Efficient Wall* costs less in energy consumption and causes less gas emissions.

6.4.3. Evaluation with the Prototype System vs. Field Test

As mentioned in Section 6.1, the energy performance of the house had been monitored before the occupants moved in the house. The data from the field tests were collected over a 21-month period from November 1993 to July 1995 (Gerbasi, 2000). Table 6-6 lists the values of energy performance derived from evaluation with the system and the field test on site.

Table 6-6. Evaluation with the System vs. Field Test

Annual Energy Consumption	Evaluation with the System	Field Test
Heating (kWh)	5814	4810
Cooling (kWh)	621	989
Appliance (kWh)	4742	5481
Total (kWh)	11177	11280

In Table 6-6, the differences of the values in the energy performance between evaluation with the system and field test are close. For example, the values of annual total energy consumption derived from evaluation and field test are 11171 kWh and 11280 kWh, respectively, where the value in evaluation is 1% less than that in field test. However, some values such as annual cooling energy consumption have significant differences. The reasons that cause these differences could be the assumptions of the HOT2000 model or the field test methods. However, in terms of approximately estimating the energy consumptions, the results from the proposed system can still be used as reference when users evaluate the energy performance of a house.

6.5. Conclusions

The NOVTEC Advanced House, which is located near Montreal, is selected as the case study house in this project because it has a set of field test data which could be used to validate the results from the implemented prototype system. In this case study, a 3D model of NOVTEC Advanced House has been created with the Architectural Desktop 2005, and an IFC text file which can represent the house has been generated by the same application. Moreover, the system imported the geometry data of the house from the IFC model, and these data can be used by the linked applications such as HOT2000 later on. By evaluating the hygrothermal and energy performance, it has been proved that the system can evaluate the energy, thermal, and moisture performance concurrently by the proposed integrated approach. Furthermore, by comparing the evaluation results with the field test data, it was shown that the results from the system are in good agreement with the field test data.

Chapter 7

Conclusions, Contributions and Future Work

7.1. Conclusions and Contributions

The research presented in this thesis attempts to establish the concept of an IFC-based framework to meet the requirements of evaluating building envelope performance. As shown in the previous chapters, such a conceptual framework has been established. Moreover, a proof-of-concept prototype system has tested and validated that this conceptual framework is rational and achievable.

The framework integrates existing simulation applications and evaluation criteria with multidisciplinary knowledge, state-of-the-art IT, and IFC. With some simplifications, a prototype system was implemented based on the concept of the proposed framework. It applies the IFC as its data model and extracts the geometry and material layer data from CAD drawings with the assistance of the Eurostep Active Toolbox. Applications such as MOIST3.0 and HOT2000 have been successfully linked to the framework. The performance values generated from the evaluation applications have been partially compared with the criteria in the evaluation tool BEPAT.

In addition, the implementation of the IFC data model in this project has shown that it makes the interoperability possible between the preprocessor and different applications during the process of the building envelope performance evaluation.

With the literature survey, different tools of computer simulation programs and evaluation criteria in different countries have been investigated and identified. It depicts the current status of the building envelope simulation and performance evaluation. This work provides not only the background of the proposed research project but also the basic knowledge for further research.

The conceptual framework presented in this thesis can facilitate the building envelope evaluation by:

(1) Simplifying the input and output data of the evaluation applications. These applications have many input and output files. By setting up default values in the input files and picking up the most useful output results in the evaluation summary, the user can use the applications with a minimum set input data, such as the weather city. Moreover, she/he can apply the most common results, such as the annual total energy consumptions.
(2) Carrying out the evaluation in short time.
(3) Operating the applications without having to learn all the details about the evaluation applications.

(4) Establishing an extensible structure. In the long term, the framework can link more applications depending on the users' needs. Moreover, besides the building envelope, it could be extended to deal with the whole building based on the current structure.

(5) Reducing the repetitive data input. Based on the same CAD model and general manual data input, the user can use the data of a building in different applications.

The proof-of-concept prototype system presented in this thesis has tested and validated the conceptual framework by:

(1) Creating a 3D and IFC model of a house and using it in the case study.

(2) Developing an interface, *Import IFC Text File*, to extract the geometry and material layer data of a house from CAD drawings in IFC data model.

(3) Developing an interface, *General Data Input*, to accommodate the data input for different simulation programs.

(4) Developing a middle layer, *Data Transfer Class*, to manage the input data in the memory for different simulation programs.

(5) Developing linkages between the framework and the performance evaluation applications, such as HOT2000 and MOIST3.0.

(6) Partially implementing the comparison between the evaluation results and a set of criteria in BEPAT.

7.2. Future Work

In order to fully exploit the capability of this IFC-based framework, further developments are recommended for implementation:

(1) The current project succeeded in extracting geometry and material layer data from IFC. The latest IFC2x2 version has integrated many data related to building services, code checking, and electrical services, which may be useful in the evaluation of the building envelope. Moreover, even though some data in the building envelope are not included in IFC2x2, it is possible to extend the IFC schema to include these data. Therefore, beside the geometry and material layer data, it is possible to extract any kind of data, such as the location of the building, which the framework needs from the IFC model.

(2) Data can be read from (extracted) and written to (stored) an IFC file in the framework. Currently, the framework can only extract data from the IFC file. In the future, it is possible to enable the framework to add new IFC objects to or update the existing objects in an IFC file.

(3) IFC is a developing internationally standard. Many extension projects are under development. It is possible to specialize an IFC extension for building envelope and include it in the future IFC release.

(4) The comparison between the criteria and the evaluation results is limited to part of BEPAT by extracting its clauses and adding them in the framework, so the automatic code compliance checking process is only at the conceptual stage. In the future, a database storing all the clauses of the BEPAT will be introduced in the framework

and the knowledge-based and code compliance checking with IFC technology may be applied to demonstrate the full concept.

(5) Link more applications. One of the significant complexities in linking to the applications is their APIs. Usually these interfaces are not provided by the vendor. This limits the number of linked applications. So far, only HOT2000 and MOIST3.0 have been linked successfully due to the availability of their APIs. As for other software, only EnergyPlus apparently has such possibility.

(6) Graphic demonstration could offer end-users with more direct explanation and easier understanding for the evaluation results. The output of the evaluation summary is in textual description so far. Graphic visualization for the output which marks the failure part of the building envelop is possible to be integrated in the framework in the future.

(7) The Internet is a convenient media to distribute software. The framework is developed as a Microsoft Windows application so far, it is possible to develop a web-based client-server project which could be distributed on the Internet and enable the end-users to use it any time and at any place.

References

ASHRAE (1989). *Fundamental Volume, ASHRAE Handbook.* Atlanta, America Society of Heating, Refrigerating and Air Conditioning Engineers.

Augenbroe, G. (2002). Trends in building simulation. *Building and Environment*, 37, 891-902.

Autodesk (2005). *Architectural Desktop 2005.* Retrieved July 18, 2005 from http://usa.autodesk.com .

Baouendi, R. (2003). *Development of a Prototype Tool for the Evaluation of Sustainability of Canadian House*, Master thesis at Concordia University, Montreal, Canada.

Baouendi, R., Zmeureanu, R., and Bradley, B. (2005). Energy and Emission Estimator: A Prototype Tool for Designing Canadian Houses. *Journal of Architectural Engineering, 11 (2)*, 50-59.

Bazjanac, V. (2004). Building energy performance simulation as part of interoperable software environments. *Building and Environment, 39*, 879-883.

Bazjanac, V. and Maile, T. (2004). IFC HVAC Interface to EnergyPlus-a Case of Expanded Interoperability for Energy. *SimBuild 2004 IBPSA-USA National Conference,* Boulder, CO, USA.

BFRL/NIST (2005). *MOIST3.0.* Retrieved July 18, 2005, from http://www.bfrl.nist.gov/863/moist.html .

Bradley, B. (2003). *BATCH HOT2000, Draft for Version 9.2.* NRCan, Canada.

BSPRO (2005). *BSPro COM-Server for IFC-Files*. Retrieved July 18, 2005 from http://www.bspro.net/ .

Buildings Group/NRCan (1995). *HOT2000 Version 7 User's Manual.* NRCan, Canada.

Buildings Group/NRCan (2005). *HOT2000.* Retrieved July 18, 2005 from http://www.buildingsgroup.nrcan.gc.ca/software/hot2000_e.html .

Burch, D., and George, T. (2001). MOIST: A numerical method for design. *ASTM MNL40*, USA.

Burch, D. and Chi, J. (1997). MOIST a PC program for predicting heat and moisture transfer in building envelopes Release 3.0. *NIST Special Publication 917*, NIST.

Citherlet, S. (2001). *Towards the Holistic Assessment of Building Performance Based on an Integrated Simulation Approach*, Ph.D. Thesis in the Architecture Department at Swiss Federal Institute of Technology (EPFL), Lausanne, Swiss.

Derome, D. (2000). *Moisture Occurrence in Roof Assemblies Containing Moisture Storing Insulation and its Impact on the Durability of Building Envelope*, Ph.D. thesis in the Department of Building, Civil, Environmental Engineering at Concordia University, Montreal, Canada.

Derome, D. and Fazio, P. (1998). Large scale testing of two flat roof assemblies insulated with cellulose. *Journal of Architectural Engineering, 6 (1)*, 12-23 .

Desmarais, G. (2000). *Impact of added insulation on the hygrothermal performance of leaky exterior wall assemblies,* Master thesis in the Department of Building, Civil, Environmental Engineering at Concordia University, Montreal, Canada.

Desmarais, G., Derome, D., and Fazio, P. (2001). Moisture performance of leaky exterior walls with added insulation, *Envelope VIII Conference*, Clearwater, Fl, ASHRAE.

Dym, C. L., and Levitt, R. E. (1999). *Knowledge-based systems in engineering.* Singapore: Mcgraw-Hill.

Eastin, I. and Boardman, P. (2000). Regulatory Change in the Japanese Residential Construction Industry, *CINTRAFOR News, 16 (2).* Retrieved July 18, 2005 from http://forests.org/archive/asia/jahosect.htm .

EOTA (2002). *Draft ETAG007.* Retrieved July 18, 2005 from http://www.cmhcschl.gc.ca/en/homadoin/excaprex/suexin/suexin_024.cfm .

ESRU (Energy Systems Research Unit) (2005). *ESP-r.* Retrieved July 18, 2005 from http://www.esru.strath.ac.uk/Programs/ESP-r.htm .

Eurostep Group (2005). *Eurostep IFC toolbox products.* Retrieved July 18, 2005 from http://www.eurostep.com/ .

Fazio, P. (1990). A case study in integrated building design. *Canadian Journal of Civil Engineering, 17 (4),* 636-642.

Fazio, P. and Poliquin, L. (2000). *Survey of Prefabricated Industry Sector – Visit of Prefabricated Companies and R&D Centers in Sweden and Finland*, April 3-8, 2000, Concordia University and Forintek Canada Corp.

Fluent Inc. (2005). *Airpak.* Retrieved July 18, 2005 from http://airpak.fluent.com/

Ge, H. (2003). *Study on overall thermal performance of metal curtain walls*, Ph.D. thesis in the Department of Building, Civil, and Environmental Engineering at Concordia University, Montreal, Canada.

Ge, H., Fazio, P. and Rao, J. (2001). Transfer of heat, moisture and air through metal curtain walls, *ICBEST 2001*, Ottawa.

Gerbasi, Dino (2000). *The energy performance of the NOVTEC advanced house*, Master thesis in the Department of Building, Civil, Environmental Engineering at Concordia University, Montreal, Canada.

GES (2005). *CONDENSE*. Retrieved July 18, 2005 from http://www.ges-int.com/en/

Goode, C, Kauffman, J. and Miller, C. L. (2002). *Beginning ASP.NET 1.0 with Visual Basic.NET*. Birmingham, UK: Wrox Press Ltd.

Gowri, K.(1990). *Knowledge-based system approach to building envelope design*, Ph.D. thesis in the Department of Building, Civil and Environmental Engineering at Concordia University, Montreal, Canada.

Granlund (2005). *RIUSKA software*. Retrieved July 18, 2005 from http://www.granlund.fi/granlund_eng/ohjelmistomyynti/frameset.htm

Graphisoft (2005). *Archicad*. Retrieved July 18, 2005 from http://www.graphisoft.com/products/archicad/?source=google_GSUS

Goldstein, B., Kemmerer, S., and Parks C. (1998). *A Brief History of Early Product Data Exchange Standard-NISTIR 6221*. NIST, U.S.A. Retrieved July 18, 2005 from http://www.mel.nist.gov/msidlibrary/doc/proddata.pdf

Gu, Qun (1999). *A knowledge-based system for energy-efficient building design*, Master thesis in the Department of Building, Civil and Environmental Engineering at Concordia University, Montreal, Canada.

Haltrecht, D., Bradley, B., and Wise, C. (1999). *Application Programmer's Interface, API specification*. NRCan, Canada.

Hammad, A., Fazio, P., and He, H.S. (2005). Computer tool to achieve better performance and integration of building envelopes with structural and mechanical systems. *The 33^{d} Annual General Conference of the Canadian Society for Civil Engineering*, Toronto, Canada.

He, H.S., Hammad, A., Fazio, P. (2005). Application of IT and international standards to improve building envelope performance. *The Ninth International Building Performance Simulation Association Conference,* Montreal, Canada.

Hensen, J. L. M., Janak, M., Kaloyanov, N. G. and Rutten, P. G. S. (1998). Introducing building energy simulation classes on the web. *ASHRAE Transactions*, 104, 1.

Horvat, M., Fazio, P., Poliquin, L., and Robichaud, F. (2001). Survey of the Canadian Prefabricated Building Industry – Findings. *International Conference on Building Envelope Systems and Technologies (ICBEST-2001)*, Ottawa, Canada.

Horvat, M., Fazio, P. and Poliquin, L. (2002). *Durability in Housing: a Review of Quality Certification Programs and Recommendations.* La durabilité dans la construction domiciliaire, programmes de certification de la qualité et recommandations, report prepared for the Société d'habitation du Québec, Québec, Canada

Horvat, M., P. Fazio, (2004), Protocol and Assessment Tool for Performance Evaluation of Light-frame Building Envelopes. *Proceedings of the CIB 2004 - World Building Congress: Building for the Future*, Toronto, 2004.

Horvat, M. and Fazio, P. (2005). Comparative Assessment of Existing Certification Programs and Assessment Tools Performance Evaluation of Residential Buildings. *Architectural Science Review journal*, University of Sydney, Sydney, Australia, 48 (1), 69-80.

Hutcheon, N.B. (1963). CBD 48: Requirement for Exterior Wall. *Canadian Building Digest*, NRC, DBR Ottawa, Canada.

Hydro Quebec (2005). *What influences your consumption?* Retrieved July 18, 2005 from http://www.hydroquebec.com/residential/ .

IAI (2005a). *A short history of the IAI and the IFC information model.* Retrieved July 18, 2005 from http://www.iai-international.org/iai_international/ .

IAI (2005b). *The EXPRESS Definition Language for IFC Development.* Retrieved July 18, 2005 from http://www.iai-international.org/iai_international/Technical_Documents/documentation/The_EXPRESS_Definition_Language_for_IFC_Development.pdf .

IAI (2005c). *Data Modelling Using EXPRESS-G for IFC Development.* Retrieved July 18, 2005 from http://www.iai-international.org/iai_international/Technical_Documents/documentation/Data_Modelling_Using_EXPRESS-G_for_IFC_Development.pdf .

IAI (2005d). *Introduction to IAI and Its IFCs.* Retrieved July 18, 2005 from http://cig.bre.co.uk/iai_uk/iai/ .

IAI (2005 e). *IFC2x edition2 model implementation guide version 1.7.* Retrieved July 18, 2005 from http://www.iai-international.org/Model/files/20040318_Ifc2x_ModelImplGuide_V1-7.pdf

IGES (2005). *About IGES.* Retrieved July 18, 2005 from http://www.nist.gov/iges/ .

Inopso GmbH (2005). *IFC2x Utility.* Retrieved July 18, 2005 from http://www.inopso.com/ .

IRC/NRC (2005). *Hygrothermal modeling tool hygIRC 1-D.* Retrieved July 18, 2005 from http://irc.nrc-cnrc.gc.ca/bes/hygIRC .

ISO 10303-11(1994). *Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 11: Description Methods: The Express Language Reference Manual.* Geneva, Switzerland: International Organization for Standardization.

ISO 10303-21 (2002). *Industrial automation systems and integration -- Product data representation and exchange -- Part 21: Implementation methods: Clear text encoding of the exchange structure.* Geneva, Switzerland: International Organization for Standardization.

ISO (1999). *International Standard ISO10303-225.* Geneva, Switzerland: International Organization for Standardization.

ISS (Integrated Structural Software) (2005). *Robot with Robin Building Modeler.* Retrieved July 18, 2005 from http://robot-structures.com/usa/products.php?PGenCategory=10 .

Karola, A., Laine, T., Lassila, K., Lahtela, H., and Jokela, M. (1999). Simulation data management in the design process. *Proceedings of Building Simulation '99, 2*, 691-695. Kyoto, Japan.

Kuenzel, H.M., Karagiozis, A.N., and Holm, A.H.A (2001). Hygrothermal Design Tool for Architects and Engineers (WUFI ORNL/IBP). *ASTM MNL40*, USA.

LEED Steering Committee (2003). *LEED Foundation Policy Manual.* U. S. Green Building Council, Washington, D.C., USA, 2003. Retrieved July 18, 2005 from http://www.usgbc.org/Docs/LEEDdocs/LEED Foundations Policy Manual - August 2003.pdf .

Liebich, T., Wix, J., Forester, J. and Qi, Z. (2002). Speeding-up building plan approvals, the Singapore e-Plan checking project offers automatic plan checking based on IFC. *The European Conferences on Product and Process Modeling (ECPPM)*, Portoroz, Slovenia.

MagiCAD (2005). *MagiCAD.* Retrieved July 18, 2005 from http://eplet.syd.kth.se/~gery/magiCAD.html .

Mund, A. T. (2002). *An evaluation framework for exterior envelope wall systems for the homebuilding industry*, Ph.D. thesis in Department of Civil and Environmental Engineering at Arizona State University, USA.

Nemetschek (2005). *Nemetschek.* Retrieved July 18, 2005 from http://www.nemetschek.com .

NRCan (2005). *R-2000 Standard.* Retrieved July 18, 2005 from http://oee.nrcan.gc.ca/r2000_standard_scope.cfm .

ORNL/IBP (2005). *WUFI ORNL/IBP V3.0.* Retrieved July 18, 2005 from http://www.ornl.gov/sci/btc/apps/moisture/ .

O' Sullivan, D.T.J., Keane, M.M., and Kelliher, D. (2002). *Specification of a STEP compliant integrated environment to support performance based assessment and control of building energy system.* Retrieved July 18, 2005 from http://www.ucc.ie/iruse/papers/osullivan-p01v1.pdf.

Quebec Agency for Energy Efficiency (2005). *Novoclimate.* Retrieved July 18, 2005 from http://www.aee.gouv.qc.ca/habitation/novoclimat/ novoclimat.jsp

Rivard, H. (1993). *CONDENSE Version 2.0 User's Manual 1st edition.* Produced by SIRICON Inc. in collaboration with the Quebec Building Envelope Council and based on the Ph.D. thesis of Gowri, K.

Romberg, R., Niggl, A., van Treeck, C., & Rank, E. (2004). Structural Analysis based on the Product Model Standard IFC. *The 10th International Conference on Computing in Civil and Building Engineering (ICCCBE),* Weimar, Germany.

SIRICON (2004). *CONDENSE User's Guide.* Version SIRICON+ (2004), GES Technologies Inc., Montreal, Quebec, Canada.

Smith, B. (1988). *The Initial Graphics Exchange Specifications.* Retrieved July 18, 2005 from http://nvl.nist.gov/pub/nistpubs/sp958-lide/246-249.pdf .

SP (2005). *SP Swedish National Testing and Research Institute.* Retrieved July 18, 2005 from http://www.sp.se/eng/spenergi/kvalitetssakring.htm .

Step Tool (2005). *What is STEP?* Retrieved July 18, 2005 from http://www.steptools.com/library/standard/ .

Suter, G. & Mahdavi, A. (2004). Elements of a representation framework for performance-based design. *Building and Environment, 39(2),* 111-234.

Timberline (2005). *Timberline Office.* Retrieved July 18, 2005 from http://www.timberline.com/software/estimating/applications/cad_integrator.aspx .

U.S. Department of Energy (2005). *EnergyPlus.* Retrieved July 18, 2005 from http://www.eere.energy.gov/buildings/EnergyPlus/ .

VTT (2005). *IFC Model Server Research and Implementation Project.* Retrieved July 18, 2005 from

http://cic.vtt.fi/projects/ifcsvr/ppt/IFCproject_en_files/frame.htm .

Wong, N. H. (2004). *Integration of CAD and Thermal Simulation through Industry Foundation Classes (IFC) Product Model*. Retrieved July 18, 2005 from http://bdg.nus.edu.sg/research/eThermal_04_017.pdf .

Appendix A: Brief Introduction of Eurostep Toolbox (Eurostep Group, 2005)

Eurostep toolboxes developed by Europ Group are recommended to implement the IFC by the IAI Implementation Support Group (IAI_ISG). There are two kinds: one is the IFC Classic Toolbox, and the other is the IFC Active Toolbox.

IFC Classic Toolbox now supports the IFC 2x schema. It is the original C++ class library toolbox. It provides pure object oriented programming methodology to access IFC data. IFC Classic Toolbox implements a unified C++ early binding interface to provide easy access to files based on the IFC schema for both reading and writing. IFC Classic Toolbox is available for all major released IFC schemas (1.5.1, 2.0 and 2x). Typical applications developed using IFC Classic Toolbox are IFC Import/Export support for AEC CAD-systems.

Eurostep Active Toolbox is an ActiveX component which provides an interface to access IFC model data. Since it applies COM technology, it is easy to incorporate into almost any application running on 32-bit MS Windows ix86 platforms. Accordingly, it can be used in VBA, Visual Basic, Delphi and different C++ IDE. So far it has several versions which comply with all major releases of the IFC schemas, such as IFC1.5.1, 2.0, 2x, and 2x2. Eurostep Active Toolbox in IFC 2x2 version has functionality such as creating IFC text files, creating IFC instances, retrieving IFC instances, deleting IFC instances, setting or editing attribute values for IFC instances, listing all attributes for an IFC instance, listing all entity names in the IFC 2x2 schema, etc.

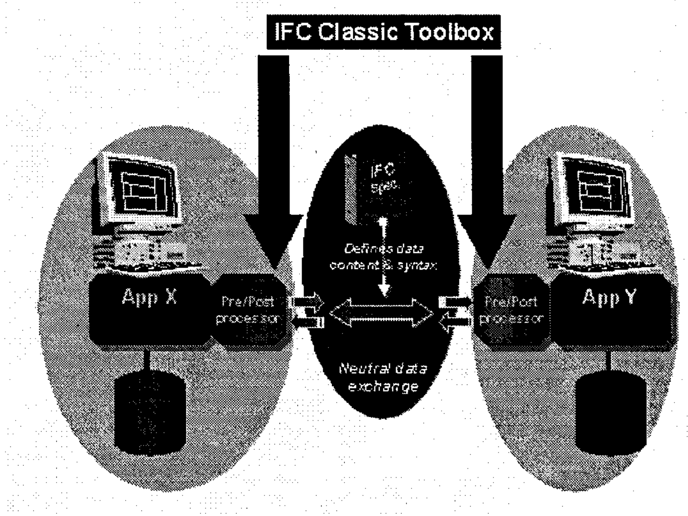


Figure A-1 Eurostep Classic Toolbox

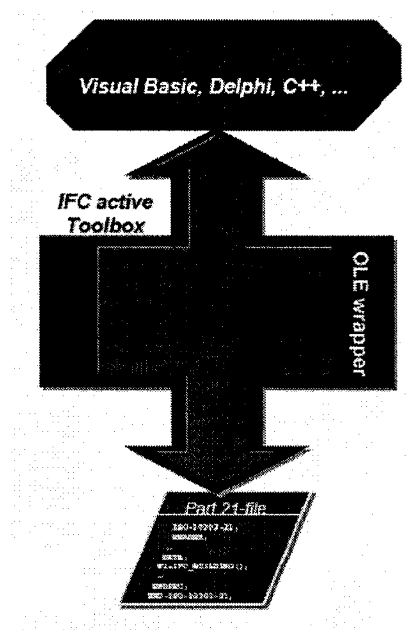


Figure A-2 Eurostep Active Toolbox

Table A-1 Functions in Eurostep Active Toolbox

Name	Function Description
readFile	Reads an IFC™/STEP data file into memory. Returns a constant denoting the file read success status.
saveFile	Saves IFC™ instances from memory to an IFC™/STEP-file. Returns a constant denoting the file save success status.
createInstance	Creates a new in-memory instance of a given entity type. Returns a handle to the created instance and a constant denoting the instance creation success status.
getInstances	Retrieves all in-memory instances of a given entity type. Returns a collection of instances of the specified entity type and a constant denoting the retrieval success status. The default behaviour is to return all the instances including all instances of all the subtypes. This can be changed to returning instances of the specified entity only using an optional selector.
deleteInstance	Deletes an instance based on a given instance handle. Returns a constant denoting the instance deletion success status.
getInstanceEntityTypeName	Retrieves the entity type name for the specified instance. Returns the corresponding entity type name and a constant denoting the retrieval success status.
getInstanceNumber	Retrieves instance index information for a specified instance. Returns the IFC™ exchange file instance number and a constant denoting the retrieval success status.
getAttributeValue	Retrieves an attribute value for a specified attribute name of a given instance. Returns the corresponding attribute value; optionally an enumeration value name (if the attribute value type is an enumeration); and optionally a select type specification (if the attribute value type is a select type); as well as a constant denoting the retrieval success status.
setAttributeValue	Sets an attribute value for a specified attribute name of a given instance. Returns a constant denoting the value assertion success status.
getATBXVersion	Retrieves version identification for the currently used Active ToolboX component. Returns the version identification string and a constant denoting the retrieval success status.
getSchemaName	Retrieves model schema identification information for the currently used version of the Active ToolboX component. Returns the schema name string and a constant denoting the retrieval success status.
getAllEntityTypes	Retrieves all entity types for the current schema. Returns a collection of the corresponding entity types and a constant denoting the retrieval success status.
getAllAttributeNames	Retrieves all attributes for a given entity type. Returns the collection of attribute names and a constant denoting the retrieval success status.
getAttributeDefinition	Retrieves attribute definitions for a given attribute of a specified entity type. Returns the attribute category (explicit, derived or inverse); an indicator for if the attribute is optional or not; an indicator for if the attribute is inverse or not; the type of attribute, data type and names; as well as a constant denoting the retrieval success status.
getEntityDefinition	Retrieves definition information for a given entity type. Returns information on if the entity type is abstract or not as well as a constant denoting the retrieval success status.
getSchemaObject	Retrieves the whole schema. Returns the schema as an object.
getEntityTypeObject	Retrieves the entity type for a specified instance. Returns the entity type as an object.

Appendix B: Default Materials in MOIST3.0

Table B-1 Default Materials in MOIST 3.0

1	AIR BARRIER (POLYOLEFIN)
2	ALUMINUM SIDING
3	BRICK
4	BUILDING PAPER, ASPHALT
5	BUILT-UP ROOFING
6	CONCRETE
7	CONCRETE BLOCK
8	FIBERBOARD, ASPHALT
9	FIBERBOARD, ROOFING
10	FIBERBOARD, SHEATHING
11	FOAM CORE SHEATHING
12	GLASS-FIBER BOARD
13	GRAVEL
14	GYPSUM BOARD
15	ICYNENE INSULATION
16	KRAFT PAPER
17	METAL ROOF DECK
18	ORIENTED STRAND BOARD
19	PARTICLE BOARD
20	PERLITE BOARD
21	PLYWOOD, EXTERIOR GRADE
22	POLYETHYLENE
23	POLYISOCYANURATE INS
24	POLYSTYRENE, EXTRUDED
25	ROOFING SHINGLES, ASPHALT
26	STRAWBALE
27	STUCCO FINISH
28	SUGAR PINE
29	VINYL SIDING
30	WAFERBOARD SIDING

Appendix C: Part of MOIST 3.0 Input Files

C.1. CONST.DAT (Building Construction Data)

```
4
S
14 0.5 70 0.5 5
N
        11   32
S
10 0.5 70 7 5
S
28 0.5 70 7 5
```

C.2. DEFSET.DAT (General Input Parameters)

```
.4579E+00   .2008E+01
.7000E+00   .9000E+02   .0000E+00   .6980E+02   .3000E+02   000000001   .9921E+01
.1915E+01
wymadwi
        000000005   .4200E+02   .7500E+02   .1000E+01
```

C.3. PDATA.DAT (Property Data of the Materials)

```
30
AIR BARRIER (POLYOLEFIN)
  .5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
  .5238E+02 .3105E+00 .9187E-01 .0000E+00 .0000E+00 .2150E+01
  .0000E+00 .0000E+00
ALUMINUM SIDING
  .1009E+00 .5356E+04 .9312E+00 .0000E+00 .0000E+00 .0000E+00
  .5240E+02 .3000E+00 .5000E-01 .0000E+00 .0000E+00 .1000E+02
  .0000E+00 .0000E+00
BRICK
  .7880E-02 .3771E+01 .8142E+00 .4867E-04 .3675E+02 .1400E+00
  .1200E+03 .1900E+00 .5170E+00 .0000E+00 .0000E+00 .3000E+01
  .0000E+00 .0000E+00
BUILDING PAPER, ASPHALT
  .5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
  .4000E+02 .3600E+00 .6800E-01 .0000E+00 .0000E+00 .3500E+00
  .0000E+00 .0000E+00
BUILT-UP ROOFING
  .5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
  .7000E+02 .3500E+00 .9400E-01 .0000E+00 .0000E+00 .3000E+00
  .0000E+00 .0000E+00
CONCRETE
  .1253E+00 .1432E+02 .8279E+00 .4917E-03 .1690E+02 .2200E+00
```

.1400E+03 .2200E+00 .1300E+01 .0000E+00 .0000E+00 .3200E+01
.0000E+00 .0000E+00
CONCRETE BLOCK
.1021E+00 .1010E+02 .7745E+00 .4917E-03 .1690E+02 .2200E+00
.5300E+02 .2200E+00 .6430E+00 .0000E+00 .0000E+00 .1920E+02
.0000E+00 .0000E+00
FIBERBOARD, ASPHALT
.1140E+01 .5060E+02 .9230E+00 .1356E-04 .3800E+01 .1680E+01
.2490E+02 .3100E+00 .3120E-01 .0000E+00 .0000E+00 .2357E+02
.0000E+00 .0000E+00
FIBERBOARD, ROOFING
.2230E+00 .6130E+01 .9010E+00 .6766E-05 .9762E+00 .1590E+01
.1879E+02 .3100E+00 .3155E-01 .3624E-04 .2618E-01 .4393E+02
.2138E+01 .2030E+01
FIBERBOARD, SHEATHING
.5305E+00 .1246E+02 .8750E+00 .6766E-05 .9762E+00 .1590E+01
.1661E+02 .3105E+00 .3170E-01 .0000E+00 .0000E+00 .2015E+02
.0000E+00 .0000E+00
FOAM CORE SHEATHING
.3640E+00 .1460E+02 .9020E+00 .0000E+00 .0000E+00 .0000E+00
.6056E+01 .3105E+00 .2802E-01 .0000E+00 .0000E+00 .2201E+01
.5926E-03 .8522E+01
GLASS-FIBER BOARD
.2630E+00 .7237E+02 .8425E+00 .0000E+00 .0000E+00 .0000E+00
.7600E+01 .1700E+00 .2083E-01 .0000E+00 .0000E+00 .1250E+03
.0000E+00 .0000E+00
GRAVEL
.1253E+00 .1432E+02 .8279E+00 .0000E+00 .0000E+00 .0000E+00
.9000E+02 .2000E+00 .2600E+00 .0000E+00 .0000E+00 .5000E+02
.0000E+00 .0000E+00
GYPSUM BOARD
.3360E-02 .1000E-07 .9010E+00 .3054E-03 .3883E+01 .1000E+01
.3900E+02 .2600E+00 .9250E-01 .0000E+00 .0000E+00 .4373E+02
.0000E+00 .0000E+00
ICYNENE INSULATION
.5554E-02 .1000E-07 .9650E+00 .0000E+00 .0000E+00 .0000E+00
.6700E+00 .2200E+00 .2310E-01 .0000E+00 .0000E+00 .4800E+02
.0000E+00 .0000E+00
KRAFT PAPER
.5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
.5240E+02 .3000E+00 .9500E-01 .0000E+00 .0000E+00 .4290E-02
.6140E-03 .4010E+01
METAL ROOF DECK
.5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
.1650E+03 .2100E+00 .1000E+03 .0000E+00 .0000E+00 .1880E-01
.0000E+00 .0000E+00
ORIENTED STRAND BOARD
.2120E+00 .3430E+01 .8110E+00 .3488E-05 .3800E+01 .1150E+01
.4000E+02 .3100E+00 .6800E-01 .0000E+00 .0000E+00 .4397E+00
.2923E-01 .4878E+01
PARTICLE BOARD
.2560E+00 .4860E+01 .8220E+00 .8796E-06 .6530E+01 .1000E+01
.4757E+02 .3105E+00 .7858E-01 .0000E+00 .0000E+00 .3420E+01
.1114E-09 .2623E+02
PERLITE BOARD
.1464E+00 .9710E+01 .8539E+00 .1550E-04 .3800E+01 .2100E+01

.1080E+02 .2600E+00 .3132E-01 .0000E+00 .0000E+00 .4379E+02
.0000E+00 .0000E+00

PLYWOOD, EXTERIOR GRADE
.3440E+00 .6180E+01 .8280E+00 .8835E-05 .2830E+01 .1330E+01
.3150E+02 .2900E+00 .4340E-01 .0000E+00 .0000E+00 .5521E+00
.1117E-02 .9765E+01

POLYETHYLENE
.5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
.5240E+02 .3000E+00 .9500E-01 .0000E+00 .0000E+00 .3600E-03
.0000E+00 .0000E+00

POLYISOCYANURATE INS.
.5280E-01 .2510E+01 .7730E+00 .0000E+00 .0000E+00 .0000E+00
.2030E+01 .2200E+00 .1213E-01 .0000E+00 .0000E+00 .1090E+01
.8018E+00 .1041E+01

POLYSTYRENE, EXTRUDED
.4194E+00 .1293E+02 .5247E+00 .0000E+00 .0000E+00 .0000E+00
.2650E+01 .2900E+00 .1670E-01 .0000E+00 .0000E+00 .1200E+01
.0000E+00 .0000E+00

ROOFING SHINGLES, ASPHALT
.5190E+02 .2538E+04 .9020E+00 .0000E+00 .0000E+00 .0000E+00
.7000E+02 .3000E+00 .1500E+00 .0000E+00 .0000E+00 .3750E-01
.0000E+00 .0000E+00

STRAWBALE
.1920E+00 .2050E+01 .7650E+00 .0000E+00 .0000E+00 .0000E+00
.1190E+02 .3900E+00 .3200E-01 .0000E+00 .0000E+00 .1200E+03
.0000E+00 .0000E+00

STUCCO FINISH
.2465E-01 .9075E+01 .9354E+00 .3054E-03 .3883E+01 .1000E+01
.4180E+02 .2600E+00 .9260E-01 .0000E+00 .0000E+00 .5000E+01
.0000E+00 .0000E+00

SUGAR PINE
.1920E+00 .2050E+01 .7650E+00 .1465E-04 .2100E+01 .1780E+01
.2279E+02 .3900E+00 .5000E-01 .0000E+00 .0000E+00 .3029E+00
.6236E-03 .9860E+01

VINYL SIDING
.1009E+00 .5356E+02 .9312E+00 .0000E+00 .0000E+00 .0000E+00
.9400E+01 .2400E+00 .3000E-01 .0000E+00 .0000E+00 .3750E-01
.0000E+00 .0000E+00

WAFERBOARD SIDING
.4600E+00 .1220E+02 .8630E+00 .1744E-05 .3800E+01 .1060E+01
.4408E+02 .2900E+00 .5252E-01 .0000E+00 .0000E+00 .3056E+00
.2255E-04 .1240E+02

Appendix D: HOT2000 Batch Version Input Files

D.1. CSV.JCF (Job Control File)

```
csv.JCC
MetSet1.v71
CsvTst.v71
```

D.2. CSV.JCC (Job Control Record)

```
.F      PARAM8.Dat    Wth910.Dir    PreScpZ      Wth910.Dat
..\Dat\                              <- Alt Path #1
                                    <- Alt Path #2
                                    <- Alt Path #3
CSVOUT   (Spreadsheet output)
  88 2 1 2 2 2 2 2 2 2 1 2 1 2 1 1 2 2 2 2 2 2  000000000000000
Builder, door, north
endselect
```

D. 3. *.V71 and *.V80 (Job Input File)

The input data for a house are stored in a pair of files, *.V71 and *.V80.

*.V71

```
46  6  0    4MMONTREAL          Walk02    612
Test v8 Walkout Type 2      Slab, No Pony Wall.          Z
Winnipeg          Manitoba          HSECHK   204-633-6363
bbradley@unies.mb.ca          Apr /02          By one finger      BcB
 1 3 1 3 1 4    0 1 1 Batch test run              2002  4  8
11M   0000102112NHCF  0 21.0 20.0  3.5S               3 1 3 2 3 3
 1  1  0  0  .400 1.000 2.790  22.0  24.0      .0      .0  176.2      .0
Ce  3  1  0000000000     00000    000000    .25000      .133     7.040
Mw  0000000000 N/A 1 4 4      2.500     48.   2.392
Do M1   2     0.000     00000      .390
Ef
Mf 0000000000     8.502    72.280      .500
S  M1  200022  0     0.000     0.000    10.000     1.800     .000     1.000
DYEN NN1177   300.0     .0     .0  3.000 14.000  3.000  4.000 234.0 2  55.0
220  50.0  50.0    .0 00.00 0000.0 .1500     .0      00.0 100.0 0      .0   00.0
3742211    10.0     3.0    .300    .500    .200   00000      .0      .0 0000000
Electric                 Baseboards 4                  2 1
A-FUELPRN3.CST
WPHHyd3   ICGWPG91  Oil1      Prop1     AvRate1
```

***.V80**

```
21      700000
 7  1  4  2  0  0  0  0  0  0  1  1
2.5     1.9     0.      0.      2.34    0.      0.      0.
0.      0.      0.      0.      0.      0.      0.      0.
0.      11.2    0.      0.      0.      0.      0.      0.
0.      0.      0.      7.11    1.      1.8
10      101     00000   1112D10   0000000000 4233004700
```

Type 2 = No pony wall, with slab H, L1, L2, L3, L4, d1, d4
C++ offsets

j =	0	Total wall height m/ft
=	1	d1 m/ft : depth of slab below grade (left , back of walkout)
=	4	d4 m/ft : " " (right, back of walkout)
=	17	L1 m/ft : length of foundation
=	27	L2 m/ft : width of foundation
=	28	L3 m/ft : distance from back to sloped grade
=	29	L4 m/ft : distance from front to sloped grade

Appendix E: Part of the IFC Text File of the NOVTEC Advanced House

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('IFC 2x'),'2;1');
FILE_NAME('F:\\He\\MasterResearch\\Thesis_He_6\\NOVTEC_House_3\\NOVTEC_House.dwg','2004-10-13T11:32:16',(''),(''),'IFC-Utility 2x for ADT V. 2, 0, 3, 5 (www.inopso.com) - IFC Toolbox Version 2.x (00/11/07)','Autodesk Architectural Desktop','');
FILE_SCHEMA(('IFC2X2_FINAL'));
ENDSEC;
DATA;
#1=IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
#2=IFCSIUNIT(*,.MASSUNIT.,$,.GRAM.);
#3=IFCDIMENSIONALEXPONENTS(1,0,0,0,0,0,0);
#4=IFCSIUNIT(*,.LENGTHUNIT.,$,.METRE.);
#5=IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.0254),#4);
#6=IFCCONVERSIONBASEDUNIT(#3,.LENGTHUNIT.,'Inch',#5);
#7=IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#8=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
...

#14=IFCGEOMETRICREPRESENTATIONCONTEXT('TestGeometricContext','TestGeometry',3,0.,#13,$);
#15=IFCPERSON('','','',$,$,$,$,$);
#16=IFCORGANIZATION('','','',$,$);
#17=IFCPERSONANDORGANIZATION(#15,#16,$);
#18=IFCAPPLICATION(#16,'IFC-Utility 2x for ADT V. 2, 0, 3, 5 (www.inopso.com)','Autodesk Architectural Desktop','');
#19=IFCOWNERHISTORY(#17,#18,$,.ADDED.,0,$,$,1113406333);
#20=IFCPROJECT('3DLbeA96P0DfrjHJ0ml3Dp',#19,'DefaultProject','Automatically generated project',$,$,$,(#14),#9);
...

#76=IFCLINE(#73,#75);
#73=IFCCARTESIANPOINT((0.,0.));
#75=IFCVECTOR(#74,124.);
#74=IFCDIRECTION((1.,0.));
#77=IFCCARTESIANPOINT((0.,0.));
#78=IFCCARTESIANPOINT((124.,0.));
#79=IFCTRIMMEDCURVE(#76,(IFCPARAMETERVALUE(0.),#77),(IFCPARAMETERVALUE(1.),#78),.T.,.CARTESIAN.);
#80=IFCSHAPEREPRESENTATION(#14,'Axis','Curve2D',(#79));
#82=IFCCARTESIANPOINT((-0.5,0.5));
...

#108=IFCCARTESIANPOINT((-8.5,0.,0.));
#109=IFCBOUNDINGBOX(#108,132.5,8.5,108.);
#110=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#109));
#81=IFCPRODUCTDEFINITIONSHAPE($,$,(#80,#106,#110));
#107=IFCWALLSTANDARDCASE('14euheRDfFQxwQGdktUE5y',#19,'','','',#72,#81,$);
#111=IFCMATERIAL('Latex Paint System');
#112=IFCMATERIALLAYER(#111,0.,.F.);

```
#113=IFCMATERIAL('GYPSUM BOARD');
#114=IFCMATERIALLAYER(#113,0.5,.F.);
#115=IFCMATERIAL('GLASS FIBER BATTS');
#116=IFCMATERIALLAYER(#115,4.,.F.);
#117=IFCMATERIAL('POLYSTYRENE, EXTRUDED');
#118=IFCMATERIALLAYER(#117,1.5,.F.);
#119=IFCMATERIAL('PLYWOOD, EXTERIOR GRADE');
#120=IFCMATERIALLAYER(#119,0.5,.F.);
#121=IFCMATERIALLAYER(#117,1.5,.F.);
#122=IFCMATERIAL('BRICK');
#123=IFCMATERIALLAYER(#122,0.5,.F.);
#124=IFCMATERIAL('No Paint No WallPaper');
#125=IFCMATERIALLAYER(#124,0.,.F.);
#126=IFCMATERIALLAYERSET((#112,#114,#116,#118,#120,#121,#123,#125),'Standard');
#127=IFCMATERIALLAYERSETUSAGE(#126,.AXIS2.,.POSITIVE.,0.);
#128=IFCRELASSOCIATESMATERIAL('1aRM88euP61x15S8YQ5IVe',#19,$,$,(#107),#127);
#129=IFCPROPERTYSINGLEVALUE('Layername',$,IFCLABEL('Wall_Storey1_In'),$);
8.5));
...

#4030=IFCCARTESIANPOINT((2.842170943040401E-013,1.13686837721616E-013,-
3.061515884555943E-016));
#4031=IFCBOUNDINGBOX(#4030,34.08869781857339,5.000000000000009,138.4126694407673);
#4032=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#4031));
#4028=IFCPRODUCTDEFINITIONSHAPE($,$,(#4027,#4032));
#4029=IFCWINDOW('1eS5KR12f5jfKELtEzPOLa',#19,'','','',#3890,#4028,$,138.4126694407673,34.0886
9781857339);
#4033=IFCRELFILLSELEMENT('16maUOXgD8_vCBw5DRPi_g',#19,$,$,#3881,#4029);
#3881=IFCOPENINGELEMENT('2hga7vyr5F8hwa6hTLpFbo',#19,'','','',#3880,#3875,$);
...

#4533=IFCCARTESIANPOINT((0.,0.,-3.061515884555943E-016));
#4534=IFCBOUNDINGBOX(#4533,40.,5.000000000000005,86.);
#4535=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#4534));
#4531=IFCPRODUCTDEFINITIONSHAPE($,$,(#4530,#4535));
#4532=IFCDOOR('1Zd_10XLL7KRqVKp83QB_3',#19,'','','',#4405,#4531,$,86.,40.);
#4536=IFCRELFILLSELEMENT('1sPiUXjlLBQQs15KPNhfip',#19,$,$,#4393,#4532);
#4537=IFCRELDEFINESBYTYPE('0AOzgler56HRCRfejFGcug',#19,$,$,(#4532),#4400);
#4393=IFCOPENINGELEMENT('1yndV3IA99U8PngjezMW4B',#19,'','','',#4392,#4387,$);
...

#5147=IFCCARTESIANPOINT((-8.5,-8.500000010000001,-1.E-008));
#5148=IFCBOUNDINGBOX(#5147,146.5,17.00000002,108.00000001);
#5149=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#5148));
#5130=IFCPRODUCTDEFINITIONSHAPE($,$,(#5129,#5145,#5149));
#5150=IFCMATERIAL('Latex Paint System');
#5151=IFCMATERIALLAYER(#5150,0.,.F.);
#5152=IFCMATERIAL('GYPSUM BOARD');
#5153=IFCMATERIALLAYER(#5152,0.5,.F.);
#5154=IFCMATERIAL('GLASS FIBER BATTS');
#5155=IFCMATERIALLAYER(#5154,4.,.F.);
#5156=IFCMATERIAL('POLYSTYRENE, EXTRUDED');
#5157=IFCMATERIALLAYER(#5156,1.5,.F.);
#5158=IFCMATERIAL('PLYWOOD, EXTERIOR GRADE');
#5159=IFCMATERIALLAYER(#5158,0.5,.F.);
#5160=IFCMATERIALLAYER(#5156,1.5,.F.);
```

```
#5161=IFCMATERIAL('BRICK');
#5162=IFCMATERIALLAYER(#5161,0.5,.F.);
#5163=IFCMATERIAL('No Paint No WallPaper');
#5164=IFCMATERIALLAYER(#5163,0.,.F.);
#5165=IFCMATERIALLAYERSET((#5151,#5153,#5155,#5157,#5159,#5160,#5162,#5164),'Standard');
#5166=IFCMATERIALLAYERSETUSAGE(#5165,.AXIS2.,.POSITIVE.,0.);
#5167=IFCRELASSOCIATESMATERIAL('2OasK2H3zEp8Nf$oUI3$DO',#19,$,$,(#5146),#5166);
...

#5754=IFCCARTESIANPOINT((4.5,-2.842170943040401E-014,0.));
#5755=IFCBOUNDINGBOX(#5754,141.5000000000001,8.500000000000028,108.);
#5756=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#5755));
#5721=IFCPRODUCTDEFINITIONSHAPE($,$,(#5720,#5752,#5756));
#5753=IFCWALLSTANDARDCASE('3QkKTBranCohh5iLQiHLiv',#19,'','Interior
Wall','',#5712,#5721,$);
#5757=IFCRELASSOCIATESMATERIAL('0hu60q4I96nPZwkk57EHis',#19,$,$,(#5753),#5562);
#5758=IFCRELCONNECTSPATHELEMENTS('1MytMN_PXDMh$tlaol8JhF',#19,$,$,$,#5753,#5549,(1,
1,1,1,1,1,1),(1,1,1,1,1,1,1),.ATPATH.,.ATSTART.);
...

#7182=IFCCARTESIANPOINT((-3.552713678800501E-014,-1.894780628693594E-014,-
3.061515884555943E-016));
#7183=IFCBOUNDINGBOX(#7182,42.,5.000000000000004,72.);
#7184=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#7183));
#7180=IFCPRODUCTDEFINITIONSHAPE($,$,(#7179,#7184));
#7181=IFCWINDOW('12vShJ3IL2Qutd0HxiLKOA',#19,'','','',#7042,#7180,$,72.,42.);
#7185=IFCRELFILLSELEMENT('0LooX2Jwr2WfFx9zTrk8H1',#19,$,$,#7036,#7181);
#7036=IFCOPENINGELEMENT('0k7LwdAwTDZwVWI3LwpkXP',#19,'','','',#7035,#7030,$);
...

#13634=IFCCARTESIANPOINT((-410.142135623731,-26.14213562373096,84.));
#13635=IFCBOUNDINGBOX(#13634,436.2842712474619,454.4725800625651,148.3275709864616);
#13636=IFCSHAPEREPRESENTATION(#14,'','BoundingBox',(#13635));
#13632=IFCPRODUCTDEFINITIONSHAPE($,$,(#13631,#13636));
#13633=IFCROOF('0u0BspwU1CvBkZIk0DkfSS',#19,'','','',#13508,#13632,$,.FREEFORM.);
...

#13901=IFCRELCONTAINEDINSPATIALSTRUCTURE('3YtykObY9B6gEIzVeSdomW',#19,$,$,(#107,
#175,#226,#276,#317,#367,#418,#473,#525,#566,#722,#784,#840,#893,#945,#987,#1058,#1110,#1179,#1
221,#1278,#1320,#1372,#1637,#1691,#1742,#1801,#1849,#1892,#2073,#2244,#2424,#2603,#2782,#2961,
#3140,#3319,#3498,#3674,#3850,#4029,#4184,#4362,#4532,#4737,#4917,#5089),#60);
#13902=IFCRELCONTAINEDINSPATIALSTRUCTURE('17dp_a8Qj1AuHGn46OvOXo',#19,$,$,(#5146
,#5215,#5267,#5309,#5361,#5403,#5445,#5506,#5549,#5605,#5653,#5695,#5753,#5815,#5858,#5901,#59
53,#5999,#6046,#6105,#6155,#6207,#6259,#6310,#6362,#6413,#6469,#6654,#6825,#7005,#7181,#7357,#
7536,#7715,#7891,#8067,#8237,#8281,#8315,#8493,#8672,#11145,#12703,#12735),#5105);
#13903=IFCRELCONTAINEDINSPATIALSTRUCTURE('2Idce3xezC$gKVVTUkkONJ',#19,$,$,(#1280
1,#12869,#12916,#12963,#13017,#13069,#13121,#13173,#13215,#13263,#13310,#13352,#13455,#13493,
#13633,#13707,#13888),#12751);
#13904=IFCRELAGGREGATES('1zYHiLpQT3meTeGaelj0pY',#19,$,$,#26,(#5105,#12751));
#13905=IFCRELAGGREGATES('1geqWurc9EN8a7BkScON9W',#19,$,$,#20,(#26));
#13906=IFCRELAGGREGATES('3GFllQ3mv9QPcU0$ywS8T3',#19,$,$,#47,(#60));
#13907=IFCRELAGGREGATES('0ak2iQ$bH8jPK_XiaDd8Rw',#19,$,$,#34,(#47));
ENDSEC;
END-ISO-10303-21;
```

Appendix F: Creating an IFC File for a Wall with the Eurostep Active Toolbox

In order to apply the Eurostep Active Toolbox to manage IFC data within a user-friendly interface, an IFC text file creation and access tool is developed. It has functionalities such as creating IFC text files, creating IFC instances, retrieving IFC instances, deleting IFC instances, setting or editing attribute values for IFC instances, listing all attributes for an IFC instance, listing all entity names in the IFC 2x2 schema, and so on. The interface of this tool is shown on Figure F-1. The sample of an IFC text file created by this tool is shown in Figure F-2. It is the *Efficient Wall* that is described in Section 6-1. The layer information is the same as listed in Table 6-3 and the cross section is also the same as drawn in Figure 6-4.

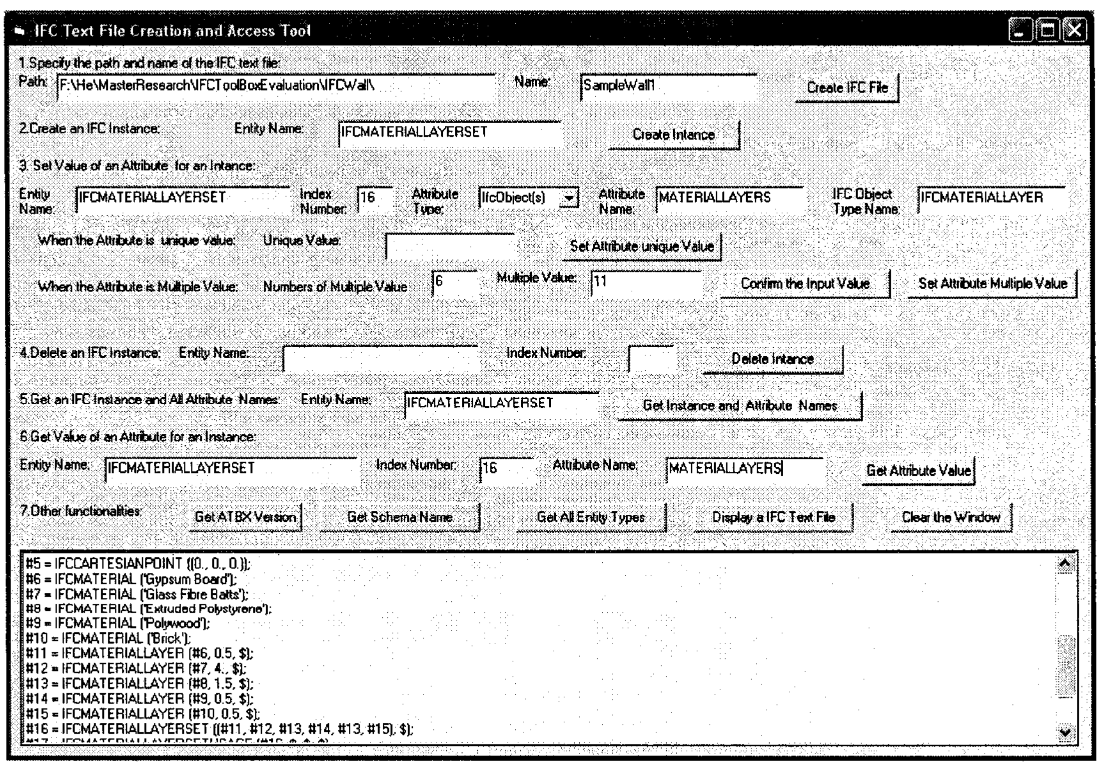


Figure F-1. IFC Text File Creation and Access Tool

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('The IFC First Wall'), '2;1');
FILE_NAME ('F:\\He\\MasterResearch\\IFCToolBoxEvaluation\\IFCWall\\SampleWall1.ifc',
'2005-03-30T11:22:21', ('Hua Sheng He'), ('Concordia University'), 'ATBX V1.0 - Ifc
Step Toolbox Version 2x2 (7. May 2003)', 'ATBX V1.0', 'BCEE & CIISE');
FILE_SCHEMA (('IFC2X2_FINAL'));
ENDSEC;
DATA;
#1 = IFCWALLSTANDARDCASE ('Global IF of the Wall', $, 'Sample Wall', 'Test', $, $, #2,
$);
#2 = IFCPRODUCTDEFINITIONSHAPE ($, $, $);
#3 = IFCSHAPEREPRESENTATION ($, $, $, $);
#4 = IFCBOUNDINGBOX (#5, 384., 8.5, 108.);
#5 = IFCCARTESIANPOINT ((0., 0., 0.));
#6 = IFCMATERIAL ('Gypsum Board');
#7 = IFCMATERIAL ('Glass Fibre Batts');
#8 = IFCMATERIAL ('Extruded Polystyrene');
#9 = IFCMATERIAL ('Polywood');
#10 = IFCMATERIAL ('Brick');
#11 = IFCMATERIALLAYER (#6, 0.5, $);
#12 = IFCMATERIALLAYER (#7, 4., $);
#13 = IFCMATERIALLAYER (#8, 1.5, $);
#14 = IFCMATERIALLAYER (#9, 0.5, $);
#15 = IFCMATERIALLAYER (#10, 0.5, $);
#16 = IFCMATERIALLAYERSET ((#11, #12, #13, #14, #13, #15), $);
#17 = IFCMATERIALLAYERSETUSAGE (#16, $, $, $);
#18 = IFCRELASSOCIATESMATERIAL ($, $, $, $, $, #17);
ENDSEC;
END-ISO-10303-21:
```

Figure F-2. An IFC Text File of the *Efficient Wall*